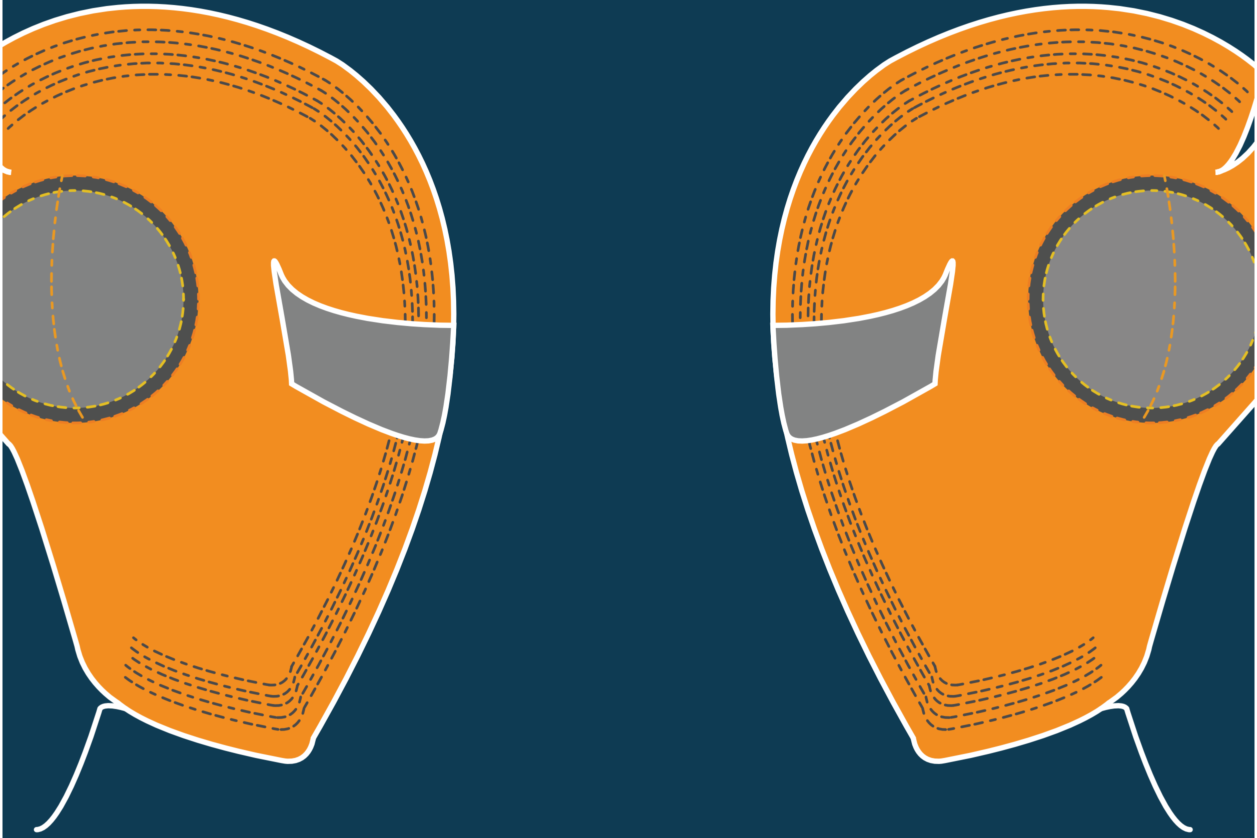
A DISSERTATION IN ARTIFICIAL INTELLIGENCE

Learning against Learning

Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions



MICHAEL KAISERS
MAASTRICHT UNIVERSITY

Learning against Learning

Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions

Dissertation

to obtain the degree of Doctor at
Maastricht University,
on the authority of the Rector Magnificus, Prof. dr. L. L. G. Soete,
in accordance with the decision of the Board of Deans,
to be defended in public
on Monday, 17 December 2012, at 16.00 hours

by

Michael Kaisers
born on 8 April 1985 in Duisburg, Germany

Supervisor: Prof. dr. G. Weiss

Co-supervisors: Dr. K. P. Tuyls
Prof. dr. S. Parsons, City University of New York, USA

Assessment Committee:
Prof. dr. ir. R. L. M. Peeters (chairman)
Dr. ir. K. Driessens
Prof. dr. M. L. Littman, Brown University, USA
Prof. dr. P. McBurney, King's College London, UK
Prof. dr. ir. J. A. La Poutré, Utrecht University
Prof. dr. ir. J. C. Scholtes

The research reported in this dissertation has been funded by a TopTalent 2008 grant of the Netherlands Organisation for Scientific Research (NWO).

SIKS Dissertation Series No. 2012-49
The research reported in this dissertation has been carried out under the auspices of the Dutch Research School for Information and Knowledge Systems (SIKS).

ISBN: 978-94-6169-331-0

Preface

This dissertation concludes a chapter of my life. It is the destination of a journey of seven years of studying, through which I grew personally and professionally. Maastricht became my second home—yet I travelled to so many other places, living in New York City, Eindhoven and New Jersey in between. When I look back and see how many wonderful people have accompanied me, I am grateful to every one of them.

Dear Karl, you have guided my studies no less than 6 years, and following my Bachelor's and Master's thesis this is the third work I have written under your supervision. I have learned a lot from you, not only scientifically. Your consideration of moral and political correctness inspires me and now influences many of my own decisions. Dear Simon, the research visit to your lab vastly broadened my horizon with respect to research, but also in a wider sense. The ideas which grew during the visit became the basis of my PhD proposal that eventually enabled me to perform four years of self-determined research, sponsored by a TopTalent 2008 grant of the Netherlands Organisation for Scientific Research (NWO). Of course, writing a dissertation is not an easy task, but your encouragements throughout the project have helped motivate me to push through. Dear Gerhard, I have enjoyed co-chairing EUMAS with you, and I appreciate the protective leadership you contribute to our department. Your fair credit assignment creates a feeling of appreciation for good work that keeps the spirit high.

Besides my promotors, my dissertation has greatly benefited from the comments and suggestions of the assessment committee. A special thanks goes to Michael Littman for his extensive comments and the inspirational time I had when visiting his lab; I still draw research ideas from the discussions we had. Dear Kurt, thank you for your comments and also for the fruitful collaboration on planning using Monte Carlo techniques, a topic that is not covered in my dissertation but that I intend to follow up on. Dear Peter, the repeated encounters in the last four years have been very encouraging, thank you for sharing your enthusiasm. Dear Ralf, Jan and Han, thank you for your assessment, I highly appreciate your support.

Many colleagues have gone part of the way with me. I enjoyed the many conversations with my colleagues in Maastricht, talking about intellectual topics, but also digressing into absurdity at times. To my friends and colleagues abroad, dear George, Tuna, Ari, and Mike, I will remember the time we had and hope our ways keep crossing in the future. I also want to express my gratitude to Jinzhong Niu and Steve Phelps for the stimulating conversations that got me off the ground in simulating auctions. To the PhD candidates who still have their dissertation ahead, it has been great to have you around, and I hope to make it back for your defenses.

A PhD should of course not be all work and no play. Luckily, my spare time was always filled with exciting events and people to hang out with. Dear Hennes, you have been great company, and professionally we became something like Karl's twins. I'm very glad we could walk the way together, and many stretches of time would have been difficult to master alone. Your smarts are incredible, and I appreciate your relaxed attitude. Dear Rena, thanks for exploring the world with me. Dear Gesa, thanks for catching me from my lows, accepting me and being honest with me. Dear Joran, Mare, Jason and Mautje, Salsa would not have been the same without you, I will never forget the many nights we danced. Dear friends of PhD Academy, I have had a great time with you, and I learned so much with you that no course could teach me, be it in the board, in one of the activities, like improvisation theater, or while organizing the PhD conference. To everyone I met while paragliding, thank you for being there with me, dreaming the dream of human flight.

While my journey would have not been half as fun without these people accompanying me, my roots have been equally important. My family has been a constant source of unconditional support, and I appreciate that deeply. *Liebe Großeltern, liebe Eltern, liebe Tanten und Onkel, liebe Cousinen und Cousins, liebe Marie, danke dafür, dass ihr einfach da seid und mir einen Hafen bietet, wenn ich von meinen Reisen zurück komme.* Last but not least, dear Pia, your love carries me through any doubt and fears. You make me believe in me, and you give me a home wherever we go.

It is impossible to name everyone who deserves mentioning; so please forgive me if your name is not here. I cherish the memories of every one who once made my day, because you all made Maastricht a beautiful place to live, and the most wonderful memory to keep.

Thank you all, *Michael*

Contents

1 Introduction

Computers have taken a prominent role in our society and automation has contributed to an unprecedented wealth in industrialized nations. The world of today is full of networks and connected systems, varying from stock exchanges and container loading to political networks [De Mesquita, 2006; Hon-Snir et al., 1998; Nouyan et al., 2009; Phelps et al., 2006]. These networks link individual humans and computers that influence each other directly or indirectly through the effect of their actions. The internet makes it is possible to connect computer systems at low cost. As a result, more and more automated systems are influencing each other directly by either communicating information or by taking actions that affect each other. Within this dissertation, each one of these individuals with an autonomous behavior is called an *agent*, and their interactions are studied by analyzing and simulating learning algorithms.

In general, one can distinguish between *software agents*, such as a simple program that can be run on a personal computer, or *hardware agents*, mostly referred to as robots. An agent can interact with its environment in two essential ways: first, it perceives the state it is in, e.g., receiving input from sensors that measure data from the real world, and second it performs actions that may change the state of the environment. A formal definition is given in Chapter 2.

Another very useful discrimination is the one between adaptive agents and static computer programs. Computer programs are more prominent and perform preprogrammed static behaviors that do not change over time. Examples of such static programs are computer applications, like a text processor, a spread-sheet application, or control software for industrial robots that simply executes a predefined sequence of actions. However, adaptive agents are taking more prominent roles as well. Consider for example an online platform that sells a commodity like airplane tickets with prices being automatically adjusted to the demand. The digital vendor is an adaptive agent that reacts to the demand it perceives from its environment. Similarly, robots may adapt to their environments. Most prominently, several vacuuming robots have arrived on the consumer market. These robots are able to adjust their behavior, e.g., route of the cleaning, according to the environment they are placed in [Choset, 2001]. In addition, advanced adaptive robots are capable to cope with far more complex environments, like the Mars rovers that explore a distant planet on our behalf with many semi-autonomous and adaptive behaviors [Barnes et al., 2006].

The need for adaptive agents is a direct consequence of challenges that arise in many domains. Pre-programmed behaviors are necessarily designed to perform well in the environment as it is seen at the time of design. However, the world changes and so do essential conditions in many applications, often in unforeseeable ways. After the change, static behaviors likely become obsolete, being inadequate or at least suboptimal. In contrast, adaptive agents are inherently more versatile and can cope with changing environments by adjusting behavior based on experience. This improvement of behavior based on experience is exactly the definition of *learning* [Sutton and Barto, 1998]. Learning becomes particularly important in the presence of several agents, because the presence of several autonomous behaviors inherently introduces elements of competition and cooperation [Stone and Veloso, 2000]. In competitive settings, the agent needs to adapt to a possibly also learning opponent. For cooperative tasks, it may seem like a plan could be devised in advance that could resolve all possible issues. However, the initial plan may be invalidated by failure of an individual agent, and if the other agents do not adapt to changes in the conditions, the system is very brittle. The true benefit of multi-agent systems is a graceful degradation if individual agents break down, and this benefit requires adaptive agents. Due to the advantages of adaptation, such as scalability and robustness, multi-agent learning is gaining popularity as a method for finding high quality behavior in very demanding environments [Panait and Luke, 2005; Shoham et al., 2007; Stone and Veloso, 2000]. It provides a distributed control mechanism for complex multi-agent systems where agents maximize their individual payoff or enact a global desired behavior while operating on local knowledge.

1.1 Motivation and scope

In essence, any agent that interacts with another learning agent is facing an environment that may change over time, i.e., it is *dynamic* rather than *static*. The optimal behavior of the agent depends on the behavior of the other agent, and if the agent seeks to behave optimally it needs to be adaptive, too. Adaptivity becomes even more essential if the agents' interests do not align and there is an element of competition and strategic behavior involved. An agent that is able to learn a better behavior by adapting to its opponent has an important competitive advantage over static agents since it can learn to exploit weaknesses of the opponent and may thereby outperform the competition.

In many real systems (e.g., business trading, sports betting and stock markets) agents are **learning against learning**, i.e., the environment faced by the learner includes other learners. This situation may lead to complex interaction patterns and the resulting system behavior is difficult to predict. Nevertheless, these systems take a central role in our society, e.g., high frequency automated equity traders account for over a third of the trading volume in the U.K. and close to three quarters in the U.S. [Department for Business Innovation and Skill, 2012]. With the rapid adoption of adaptive technology in critical positions with wide-ranging effects, there is a growing need for a deeper understanding of the dynamics of multi-agent learning. Such insights are essential for the prediction of system stability, the regulation of systems like markets, and the development of agents that perform well in cooperation or competition with each other.

The focus of this dissertation is on the effect that learning has on the learning processes it interacts with. This effect is elicited by studying learning algorithms in strategic interactions. Depending on the learning algorithms involved, the joint behavior may settle such that no agent has an incentive to adapt further, or the agents may continue to adapt to each other for eternity. This dissertation studies the qualitative long-term behavior of multi-agent reinforcement-learning algorithms, i.e., algorithms that learn from trial-and-error interactions. Using algorithmic models of learning makes large numbers of repeatable experiments possible under very controlled settings on the one hand, and a formal theoretical analysis of the interactions on the other hand. These two complementary approaches yield deep insights into the dynamics of interactive learning behavior, and throughout the dissertation analytical insights will be illustrated with simulation experiments. The insights into these algorithmic learning processes may be taken as a model of human learning, e.g., the algorithm Cross Learning (formally introduced in Section 2.2.1) has been devised to match data of human learning [Cross, 1973]. In addition, the increased adoption of adaptive agents makes their understanding crucial for well-behaved real systems, e.g., on May 6, 2010 the Dow Jones industrial average dropped more than 600 points within minutes, and then recovered rapidly [Bowley, 2010]. This event, known as the *Flash Crash* of 2010, has been attributed to the impact of autonomous trading and its cascading effects [Bowley, 2010; Lauriciella et al., 2010]. The prominent role of learning agents, e.g., in equity and futures contracts trading, makes the study of *learning algorithms* in strategic interactions not only a model of human learning but also an engineering goal in itself.

Multi-agent learning is a challenging problem and has recently attracted increased attention by the research community [Busoniu et al., 2008; Shoham et al., 2007; Stone, 2007; Tuyls and Parsons, 2007]. Learning in multi-agent environments is significantly more complex than single-agent learning, since the optimal behavior to learn depends on other agents' policies. These policies are in turn changed according to the other agents' learning strategies, which makes the first agent's learning goal a moving target. All agents face this situation, while chasing their own dynamic learning goal they directly influence and move the learning goals of other agents. This makes predicting the behavior of learning algorithms in multi-agent systems difficult. More formally, the environment may be in a set of states, and the state transitions are influenced by the choices of all agents. Thus, each agent faces a non-stationary environment in which the Markov property does not hold, i.e., information available to the agent does not fully specify the state transition probabilities, because they depend on the concealed policies of the opponents. Unfortunately, many proofs of convergence to optimal policies in the single-agent learning literature depend on the Markov property and thus become inapplicable in multi-agent settings. This limits the theoretical backbone available for multi-agent learning. In contrast to single-agent learning, which has been widely studied both experimentally [Van den Herik et al., 2007] and theoretically [Auer, 2002; Kaelbling et al., 1996; Watkins and Dayan, 1992], the understanding of multi-agent learning is still rather immature [Shoham et al., 2007; Stone, 2007].

1.2 Related work

This section gives a brief outline of the different streams of related work. A far more comprehensive overview is presented in Chapter 2.

Strategic interaction of several autonomous agents is the classical subject of game theory, which captures the strategic conflict of interests formally in a *game* [Gibbons, 1992]. A game has a number of players and each player has a number of strategies to choose from. In addition, each player has a payoff function over the outcomes of the game, which assigns a numerical payoff value to the desirability of each possible strategy constellation. The focus of classical game theory is to elicit strategic properties that are inherent to the game. It is assumed that all players are capable and willing to compute and enact their best possible strategy — one that maximizes the players' payoff values given the information available to them. This assumption is called *perfect rationality*. Rationality and the focus on game properties rather than players is central to classical game theory; as a corollary, classical game theory is less concerned with the process of how players find their strategies. In contrast, multi-agent learning is primarily concerned with how players reach good strategies. In the analysis of learning behavior, some game properties are used to relate the learning behavior to rationality or optimality.

Multi-agent learning survey papers and publications at agents and machine learning conferences make clear that the number of multi-agent learning algorithms to choose from is constantly growing [Abdallah and Lesser, 2008; Blum and Mansour, 2007; Busoniu et al., 2008; Hu and Wellman, 2003; Panait and Luke, 2005]. Many domain-specific problems are tackled by modifying or refining the learning algorithms in question for the task at hand. An overview of well-established multi-agent learning algorithms with their various purposes is given in [Busoniu et al., 2008], which demonstrates the need for a comprehensive understanding of their similarities and differences. The diversity of learning algorithms makes it imperative to specify the assumptions (*learning bias*) [Crandall et al., 2011]. These assumptions are particularly diverse with respect to what information each agent observes, be it only their own reward or also including other agents' actions and possibly rewards. Also, the full payoff function may be available to an agent ahead of playing the game. In this work, the agents only observe their own payoffs and are able to remember the actions that were taken. The payoff function is not available ahead of the game, since that is hardly ever the case in realistic applications. Neither does the agent observe the number of opponents or the actions taken by them. This makes reinforcement learning an applicable model for a variety of task domains [Sutton and Barto, 1998; Tuyls and Parsons, 2007]. Reinforcement learning seeks successful behavior through trial-and-error interactions with the environment. However, contemporary reinforcement-learning algorithms often feature a number of parameters that require tuning, a cumbersome task.

Evolutionary game theory has been linked to reinforcement learning and provides useful insights into learning dynamics [Börgers and Sarin, 1997; Gintis, 2009; Tuyls et al., 2006, 2003]. In particular, this link has provided insights into the dynamics and convergence properties of current state-of-the-art multi-agent reinforcement-learning algorithms such as Q-learning [Wunder et al., 2010]. It makes it possible to study the resilience of equilibria, visualize the basins of attraction and fine tune parameters.

This dissertation studies multi-agent learning dynamics formally and is based on two branches of literature that 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*. Notable algorithms in this branch include Infinitesimal Gradient Ascent (IGA) [Singh et al., 2000], the variation Win or Learn Fast IGA (WoLF) [Bowling and Veloso, 2002] and the Weighted Policy Learner [Abdallah and Lesser, 2008]. The second branch is concerned with learning in unknown environments based on Reinforcement Learning. In this case, the learning agent updates its policy based on a sequence of $\langle \mathtt{action}, \mathtt{reward} \rangle$ pairs that indicate the quality of the actions taken. Notable algorithms include Cross Learning [Cross, 1973], Regret Minimization [Klos et al., 2010], and variations of Q-learning [Kaisers and Tuyls, 2010; Watkins and Dayan, 1992].

Previous work has established that Cross Learning, which implements a simple learning automaton, converges to the replicator dynamics from evolutionary game theory as the learning update steps become infinitesimally small [Börgers and Sarin, 1997]. The replicator dynamics have also been recognized at the heart of Regret Minimization [Klos et al., 2010]. In addition, one of the most popular reinforcement-learning algorithms, namely Q-learning, has been decomposed into exploration terms that encode information gain, and exploitation terms that are equivalent to the replicator dynamics [Tuyls et al., 2006]. In other words, Q-learning follows dynamics similar to Cross Learning and Regret Minimization but enriched with exploration. It should be noted that these initial results for Q-learning were derived from the simplifying assumption that Q-learning would update all actions at every iteration. These dynamics will therefore be referred to as the *idealized* model of Q-learning.

1.3 Problem statement

Experiments comparing Q-learning to its idealized evolutionary model reveal two interesting facts: one, the learning trajectories deviate significantly from the predicted dynamics, and two, the idealized learning dynamics are more desirable than the actual learning behavior. More specifically, the behavior of Q-learning varies depending on the initialization and may temporarily decrease the probability of playing clearly superior strategies during the learning process[1]. In contrast, the idealized model prescribes trajectories that monotonically increase the probability for playing game theoretically preferable actions. A detailed elaboration of the causes for the mismatch is lacking. Furthermore, the inconsistent behavior of Q-learning has made it difficult to analyze the long-term behavior in multi-agent settings. It is not proven yet whether individual Q-learning applied to multi-agent settings would in the long run stabilize to some fixed behavior, cycle in a repeating pattern or even form chaotic behavior. It is conjectured but proven that the long term behavior is related to the core solution concept of classical game theory, namely Nash equilibria.

The evolutionary models for multi-agent learning are so-far only established for a limited number of algorithms in single-state games. In fact, specific assumptions have been made to simplify derivations, e.g., Q-learning has been assumed to update all actions at every step, and its exploration parameter has been assumed constant. These assumptions conflict with single-agent learning convergence theory that suggests decreasing exploration over time in order to find the global rather than a local optimum [Watkins and Dayan, 1992]. In addition, real applications can seldom be modeled as single-state games and more naturally relate to multi-state games. These limitations should be alleviated to make the methodology more applicable to realistic problems.

Tuning time-dependent parameters, such as exploration of learning algorithms, remains a cumbersome task even given an available evolutionary model. Although the state-of-the-art techniques are well-suited to study force fields that are constant over time [Bloembergen et al., 2011; Tuyls et al., 2006], they have not been designed for the study of learning dynamics that change over time. The trajectory plots and directional field plots commonly used in the literature are a good basis, but they lack an essential time dimension. Hence, there is a need for a tool to systematically design and analyze time-dependent parameters.

As indicated in the previous section, the literature is divided between the study of *infinitesimal gradient ascent* and *replicator dynamics*. However, both streams of literature appear to pursue a common goal: relating convergence of a dynamical system to multi-agent learning interactions. However, the commonalities and differences have not been discussed explicitly yet; it is a gap in the literature that needs to be closed.

The application of an evolutionary analysis to complex real world problems was pioneered by Walsh et al. [Walsh et al., 2002] who introduced a systematic way to capture payoffs from practical domains. These initial experiments used simple dynamics, but the method is general and any dynamics can be used in conjunction with the payoffs gathered in practical domains. In particular, it is worth investigating how exploration affects the qualitative results of the model.

[1] The decrease in probability for playing dominating strategies may commonly be transient, but examples can be constructed such that it lasts for an arbitrarily long time span (see pessimistic initializations in Section 3.2.2).

In sum, interacting agents are ubiquitous nowadays, and these agents in strategic interactions can be modeled and controlled by multi-agent learning. *Learning against learning* produces systems with highly complex dynamics where the emergent collective behavior is difficult to predict, and our understanding of their stability is limited. Nevertheless, such systems (e.g., stock markets) are central to modern societies and have large stakes at risk not only for individuals but for society at large. This dissertation tackles this fundamental gap and contributes to the theoretical framework for the analysis of multi-agent learning.

1.4 Research questions

The following research questions were distilled from the problem statement. They all address the problem of how to use and improve the evolutionary framework for the analysis of reinforcement-learning algorithms in strategic interactions.

1. Why does Q-learning deviate from the idealized model, and how can Q-learning be adjusted to show the preferable behavior of the idealized model? *Chapter 3*
2. What is the long term behavior of this idealized Q-learning model; does Q-learning converge to Nash equilibria? *Chapter 3*
3. How can the evolutionary framework be extended to more realistic scenarios such as varying exploration rates or multiple states? *Chapter 4*
4. Are there alternative perspectives on the time-varying dynamics of multi-agent learning that enable a systematic design of time-dependent parameters? *Chapter 5*
5. What are the commonalities and differences between variations of infinitesimal gradient ascent and the replicator dynamics? *Chapter 5*
6. How can the evolutionary analysis be applied in realistic domains, and more specifically what does it reveal about auctions and poker? *Chapter 6*

Each of these research questions is addressed in the chapter that is indicated in italics. An explicit answer to each research questions is given in Section 7.1. In answering these questions, this dissertation supports the thesis that deriving and then scrutinizing dynamical systems of multi-agent reinforcement learning provides valuable insights into their strategic interactions.

1.5 Contributions and structure of this dissertation

The chapters of this dissertation can be grouped into three parts: Chapters 1 and 2 provide an introduction and required background knowledge, Chapters 3–6 give a detailed account of the research contributions, and Chapter 7 concludes with a discussion of the methodology and answers to each research question.

A survey of state-of-the-art methods for analyzing multi-agent reinforcement learning and their limitations is given in Chapter 2. They range from empirical competitions in benchmark problems to analyzing learning dynamics theoretically to determine convergence behavior and deliver performance bounds. Theoretical advances have been largely due to a dynamical systems approach that links multi-agent reinforcement learning to evolutionary game theory. However, this framework has several important limitations that need to be addressed before applying it to real-world problems. (1) The idealized dynamical model of Q-learning deviates from the average algorithm behavior, and it assumes a constant exploration rate which conflicts with best practice. (2) Few algorithms have been described and can be analyzed in this framework. (3) Much of the literature solely considers single-state environments. (4) The literature is divided between a discussion of gradient-based and feedback-based algorithms. Each of these limitations is addressed in this dissertation, and several other insights add to the literature. Overall, the contributions of this dissertation can be summarized as follows:

The first contribution is an in-depth analysis of the discrepancy between the average behavior of Q-learning and its idealized dynamical model assuming simultaneous updates of all actions. Results show that the dynamical system features more rational learning trajectories than the average behavior of Q-learning. For that reason, I derive and propose the algorithm Frequency Adjusted Q-learning (FAQ-learning) that inherits the convergence behavior of the formal model. FAQ-learning is introduced, evaluated and analyzed in Chapter 3.

The second contribution is a proof of convergence for FAQ-learning in two-action two-player games, constructed within the evolutionary framework. FAQ-learning converges to stable points, which for low exploration move close to Nash equilibria. In Battle-of-Sexes type games, a bifurcation of attractors occurs at a critical exploration rate. This proof of convergence and discussion of FAQ-learning dynamics concludes Chapter 3.

Contribution number three is an extension of the dynamical systems methodology to more realistic settings. (1) The model of FAQ-learning is extended to cover time-dependent exploration rates. (2) Several options for extending the framework to multi-state environments are discussed. (3) A lenient variant[2] of FAQ-learning is derived that increases the probability to converge to the global optimum in cooperation games. These three extensions to the dynamical systems framework are presented in Chapter 4.

Fourth and fifth, two new perspectives on multi-agent learning dynamics are introduced: (1) An orthogonal view complements existing analysis, especially when designing time-dependent parameters, (2) The bipartite multi-agent learning literature (focus on evolutionary vs. gradient ascent dynamics) is unified by proving that multi-agent reinforcement learning implements on-policy stochastic gradient ascent. These new perspectives are presented in Chapter 5 and pave the way for further cross-fertilization between gradient ascent literature and evolutionary game theory.

Sixth, the viability of this framework is demonstrated by analyzing heuristic strategies in auctions and poker. In both domains, the analysis provides additional

[2] The underlying algorithm is inspired by the idea of forgiving mistakes of other players to coordinate.

insights in support of expert knowledge. Previous research has shown that the value of information in auctions may counter-intuitively be not monotonic but rather follow a J-curve. This finding is confirmed by a variety of more realistic simulations. Results show that the cost of information has a significant impact on the survival of lower informed traders and their continuing presence in a trading population. The analysis of poker strategies confirms expert advice that predicts aggressive strategies to dominate their passive counterparts. The analysis of data from human poker games reveals that expert advice matches most closely to models that include exploration. Both case studies, auctions and poker, are presented in Chapter 6.

Finally, the dissertation is concluded in Chapter 7. This last chapter discusses the findings in relation to other multi-agent learning research, and it provides explicit answers to the research questions. Some limitations of the deployed methodology are pointed out and provide a basis for future research.

The dissertation has been organized such that common terminology and required concepts are introduced and formally defined in Chapter 2. However, the contribution of chapters (Chapter 3–6) may be read individually by the domain expert. Chapter 4 builds on the algorithm introduced in Chaper 3. Chapters 5 and 6 do not build on but rather complement the other contribution chapters. The interested reader may thus choose to either start with an introduction to previous research and open challenges in Chapter 2, or jump right in with any one of the contribution chapters and refer back to the formal definitions of Chapter 2 only where necessary.

1.6 Relation to published work

The background knowledge presented in Chapter 2 is based on work of other authors and cites many relevant sources from literature. The contribution chapters (Chapter 3–6) are mostly based on work that has already been published at peer-reviewed conferences, workshops or journals. Chapter 3 is based on two publications; the first constitutes Section 3.1–3.2, and the second yields the arguments of Section 3.3:

Section 3.1–3.2 Michael Kaisers and Karl Tuyls. Frequency Adjusted Multi-agent Q-learning. In van der Hoek, Kamina, Lespérance, Luck, and Sen, editors, *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315. International Foundation for AAMAS, 2010.

Section 3.3 Michael Kaisers and Karl Tuyls. FAQ-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), 2011.

Chapter 4 comprises three extensions to the evolutionary framework. The arguments given in Section 4.1 and 4.3 have been published while Section 4.2 presents several extensions to multi-state games of which only Section 4.2.4 is published.

Section 4.1 Michael Kaisers, Karl Tuyls, and Simon Parsons. An Evolutionary Model of Multi-agent Learning with a Varying Exploration Rate (Extended Abstract).

In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1255–1256. International Foundation for AAMAS, 2009.

Section 4.2.4 Daniel Hennes, Michael Kaisers, and Karl Tuyls. RESQ-learning in stochastic games. In *Adaptive and Learning Agents (ALA 2010) Workshop*, 2010.

Section 4.3 Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Empirical and Theoretical Support for Lenient Learning (Extended Abstract). In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1105–1106. International Foundation for AAMAS, 2011.

Chapter 5 presents two new perspectives that have been proposed in the following publications.

Section 5.1 Michael Kaisers. Replicator Dynamics for Multi-agent Learning - An Orthogonal Approach. In Toon Calders, Karl Tuyls, and Mykola Pechenizkiy, editors, *Proc. of the 21st Benelux Conference on Artificial Intelligence (BNAIC 2009)*, pages 113–120, Eindhoven, 2009.

Section 5.2 Michael Kaisers, Daan Bloembergen, and Karl Tuyls. A Common Gradient in Multi-agent Reinforcement Learning (Extended Abstract). In Conitzer, Winikoff, Padgham, and van der Hoek, editors, *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1393–1394. International Foundation for AAMAS, 2012.

Section 5.2 Michael Kaisers and Karl Tuyls. Multi-agent Learning and the Reinforcement Gradient. In Massimo Cossentino, Michael Kaisers, Karl Tuyls, and Gerhard Weiss, editors, *Multi-Agent Systems. 9th European Workshop, EUMAS 2011*, pages 145–159. Lecture Notes in Computer Science, Vol. 7541. Springer, 2012.

Chapter 6 contains the analysis of strategies in the two application domains *double auctions* and *poker*. Both parts are based on a publication.

Section 6.1 Daniel Hennes, Daan Bloembergen, Michael Kaisers, Karl Tuyls, and Simon Parsons. Evolutionary Advantage of Foresight in Markets. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 943–949, 2012.

Section 6.2 Marc Ponsen, Karl Tuyls, Michael Kaisers, and Jan Ramon. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1 (1):39–45, January 2009.

A full list of my publications is given at the end of my dissertation (see page 145).

2 Background

This chapter introduces concepts from reinforcement learning, game theory, and dynamical systems that form the basis of the theoretical framework used throughout this dissertation. First, the core concepts of reinforcement learning are introduced, and reinforcement-learning algorithms are presented. Next, the challenges of applying and analyzing these algorithms in multi-agent settings are discussed. Game theory provides a framework to capture strategic conflicts, and dynamical systems provides a grasp on the interactive dynamics. Subsequently, the link between reinforcement learning and dynamical systems is explained, which ties reinforcement-learning algorithms, learning dynamics and game theory together. Finally, limitations of state-of-the-art approaches are pointed out and serve as a departing point for the contributions of this dissertation.

2.1 Reinforcement learning

Reinforcement learning is applicable in environments where behaviors can be evaluated by trying them, but there may be no or little *a priori* guidance of what constitutes good behavior. Reinforcement learning is based on a simple reward signal given as a response to the sequence of actions that the agent executes. This attribute sets it apart from other learning paradigms like supervised learning, which requires an explicit specification of the desired outputs for a set of sample inputs (actions), or unsupervised learning, which has neither an error nor a reward signal to evaluate potential solutions [Sutton and Barto, 1998]. The existence of a numeric quality score makes it closest to the field of evolutionary computation, a link that is made explicit by formal derivations in Section 2.5.

Consider a simple reinforcement-learning task that some readers may be familiar with: Suppose you would like your dog to learn to fetch the newspaper. The dog does not understand any explicit commands, but it does act autonomously and is bound to try all kinds of things. Whenever the dog shows the desired behavior you can reward it to reinforce this behavior. It usually drastically speeds up the learning process if reward for partial achievement of the desired result is given. In this example, the human specifies the feedback signal for the dog, and the dog is the learning agent. It is possible that the dog first needs to observe the state of the environment, e.g., whether there is a newspaper or not. In this case, the reward may be conditional on the state and action, and different behavior may be rewarded depending on the state. Figure 2.1 depicts the agent-environment interaction schematically. The learner relates the reward signal to previously executed actions to learn a behavior that maximizes cumulative future reward [Sutton and Barto, 1998].

Several repetitions by the trainer are necessary before the optimal behavior is shown consistently by the learner. This observation is not a flaw of reinforcement learning, but rather inherent to the exploration of alternative actions that might, as long as unexplored, still yield unknown higher rewards. In addition, the reward feedback may be stochastic and several samples of the same action may be necessary to estimate the expected return. The agent must balance exploiting what is known about good actions with exploring other actions that have not yet been tried at all or actions where the expected payoff is still uncertain. This active sampling process has been a paradigm shift from consulting statistic experts after experiments have been carried out to an online *exploration-exploitation tradeoff* [Robbins, 1952]. This tradeoff demands to balance performing close to optimal with respect to the information that has been collected in previous interactions with improving certainty of what is good.

2.1.1 The multi-armed bandit problem and regret

The multi-armed bandit problem is a formalization of a basic one-state reinforcement-learning task. Consider an agent walking into a casino that contains several one-armed bandits. The agent may choose at each time step $\mathbf{t}$ which bandit to draw from, where

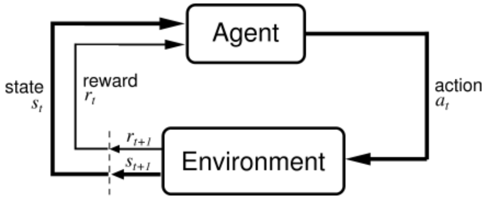


Figure 2.1: A schematic depiction of the agent-environment interaction: a state-dependent feedback is given as a response to the executed action. This illustration is adopted from published work [Sutton and Barto, 1998].

time may be of infinite horizon or finite horizon, i.e., $t \in \{1, 2, \ldots, t_{max}\}$ with t_{max} some finite integer or $t \in \mathbb{N}$. Consider the infinite horizon problem. Each bandit $i \in A$ returns a stochastic payoff R_i according to an unknown underlying distribution. In the basic model the distributions of rewards are stationary for all arms, i.e., they do not change over time. The agent seeks to maximize the payoff that it receives, and thus aims at drawing the bandit with the highest expected return [Weber, 1992]. An elaborate discussion of variations of this problem is given elsewhere [Gittins et al., 2011].

The behavior of the agent is described by the policy $\pi : \mathbb{N} \to A$, where $\pi(t) = i$ assigns action i to time step t. The agent seeks to find the policy π that maximizes his cumulative future reward. The cumulative future reward is taken as the discounted sum of returns, where γ is the discount factor with $0 \leqslant \gamma \leqslant 1$:

$$\sum_{t=0}^{\infty} \gamma^t R_{\pi(t)}.$$

This model subsumes the undiscounted infinite horizon problem with $\gamma = 1$. While discounting is optional in finite-horizon problems, not discounting leads to the following problem in infinite-horizon problems: two policies that play optimally from different points in time give rise to the same limit reward, since the optimal reward becomes dominant and the sum unbounded. Nevertheless, the policy that arrives at optimal behavior earlier is preferable. For $0 < \gamma < 1$, the sum of rewards is bounded and the value of γ can be chosen to tune the optimization from far-sighted to myopic. If $\gamma = 0$, the agent only optimizes the immediate reward. This discrimination becomes particularly important in multi-state optimization, as described in the following section.

Whenever the agent draws an arm other than the best one it incurs a certain **regret**. Formally, regret is defined as the difference between the obtained payoff and the maximal expected payoff that could have been achieved by behaving optimally right from the start, i.e., pulling the arm with the highest expected value $R^* = \max_i E[R_i]$ every time. Thus, in an auxiliary formulation to maximizing the cumulative reward, the agent tries to minimize its regret. Formal regret bounds have been one driving factor for the development of new reinforcement-learning algorithms for multi-armed bandit type problems [Agraval, 1995; Audibert et al., 2010; Auer, 2002; Auer et al., 2002; Jaksch et al., 2010; Kaelbling et al., 1996]. However, these regret bounds are hard to obtain [Kaelbling et al., 1996]. In multi-agent settings, reward distributions become non-stationary from the agent's point of view and developing a meaningful notion of regret becomes difficult.

In practice, it may be sufficient to find a behavior that is almost as good as the optimal behavior. This concept is formally developed in **Probably Approximately Correct** (PAC) learning [Valiant, 1984]. It answers the following question: How many interactions are necessary to find ϵ-optimal behavior with probability of at least $1 - \delta$ [Even-dar et al., 2002; Kaelbling et al., 1996]. PAC bounds are available for some algorithms in multi-armed bandit problems [Even-dar et al., 2002] and in Markov decision processes [Strehl et al., 2006].

Nowadays, insights into the multi-armed bandit problem find wide application in online advertisement. Ad engines need to select an ad to display to a specific user embedded in a website. The user behavior, i.e., clicking an ad or not, provides feedback to the ad engine, which seeks to display the ads with the highest revenues. In practice, many other constraints need to be taken into account, such as the limit in ad placement budgets purchased by clients from the ad engine, and the fact that a user navigates through websites and the choice of ads on a previous page may influence the expected payoffs for ads on following pages. These limitations provide a further incentive for the ad engine to explore alternative ads, and to model the problem as a multi-step optimization problem. The following section presents a framework for such a multi-step optimization.

2.1.2 Markov decision processes

Markov decision processes (MDPs) provide a formal model of agent-environment interactions in which the outcome is influenced by both stochastic influences and the actions of the agent [Howard, 1960; Puterman, 1994]. They describe discrete time stochastic control processes, where at each time step t, the process is in some state s, and the agent chooses to play action i of the available actions in state s. In response to this action, the process stochastically moves to a new state s' with probability $T_i(s, s')$, and provides a reward of $R_i(s, s')$ to the agent. Thus, a Markov decision process is a 4-tuple $(S, A(\cdot), T_{\cdot}(\cdot,\cdot), R_{\cdot}(\cdot,\cdot))$, where S is a set of states, $A(s)$ is the set of actions available in state s, $T_i(s, s')$ specifies the transition probability, i.e., the probability of moving from s to s' after selecting action i, and $R_i(s, s')$ denotes the reward given after selecting action i in state s and successfully moving to state s'. The multi-armed bandit problem presented in the previous section is the special case of a one-state Markov decision process.

It is worth noting that $T_i(s, s')$ is independent of any previous states, i.e., the stochastic state transitions of the process solely depend on the current state, which is also called the **Markov property**. As a consequence of this fact, the optimal policy for this problem can be written as a function of the state only. The behavior of the agent is described by the policy $\pi : S \to A$, where $\pi(s) = i$ assigns action i to state s. The agent seeks to maximize the discounted sum of rewards, where $0 \leqslant \gamma \leqslant 1$ is the discount factor:

$$\sum_{t=0}^{\infty} \gamma^t R_{\pi(s_t)}(s_t, s_{t+1}).$$

The **value function** $V^\pi(s)$ denotes the value of being in state s and executing policy π. It can be expressed recursively in relation to the values of all other states s':

$$V^\pi(s) = \sum_{s'} T_{\pi(s)}(s, s')\big[R_i(s, s') + \gamma V^\pi(s')\big].$$

The optimal value function $V^*(s) = \max_\pi V^\pi(s)$ indicates the value of the state given the optimal policy. The optimal policy $\pi^*(s)$ chooses the action with the maximal expected reward at each state:

$$\pi^*(s) = \arg\max_i \left\{ \sum_{s'} T_i(s,s') \left[R_i(s,s') + \gamma V^*(s') \right] \right\}. \tag{2.1}$$

If the Markov decision process is known, i.e., the transition and reward functions are given, two main approaches can be used to compute the optimal policy: Value iteration or policy iteration. **Value iteration** incrementally improves estimates of the value function using the *Bellman equation* [Bellman, 1957]:

$$V(s) \leftarrow \max_i \left\{ \sum_{s'} T_i(s,s') \left[R_i(s,s') + \gamma V(s') \right] \right\}.$$

This update rule is repeated for all states until convergence, i.e., until the left hand side equals the right hand side up to a marginal error. Once the optimal value function is found, the optimal policy can be computed from it using Equation 2.1. **Policy iteration** maintains an estimate of a good policy and directly updates it for incremental improvements [Howard, 1960]. However, since the reward and transition functions are usually not known to the agent [Sutton and Barto, 1998], neither approach is discussed in detail here. Section 2.2 describes reinforcement-learning algorithms that find good policies without having direct access to the transition and reward functions.

2.1.3 Stochastic games

Stochastic games provide a model for strategic interactions of several players. They generalize Markov decision processes to multiple agents and extend repeated games to multiple states [Littman, 1994; Neyman, 2003]. First, consider a repeated **normal form game** as a special case of a one-state stochastic game. A normal form game is defined by the tuple (N, A, R). All players $p \in N$ simultaneously have to choose from their set of available actions A_p, where $A = A_1 \times \ldots \times A_n$. Each player p has a payoff function $R^p_{\vec{a}}$ over the joint actions $\vec{a} = (a_1, \ldots, a_n)$, where a_p denotes the action chosen by player p. For two-player games, the payoff function can be given in a bi-matrix form. Normal form games are the subject of classical game theory and an example is given and discussed in Section 2.4.1. If the players encounter each other several times in the same normal form game it is called a **repeated game**.

A **stochastic game** is a stochastic process that moves between several states, and the players play a specific normal form game in each state. It is defined by the tuple (S, N, A, T, R), where

- S is a set of states
- N is a set of n players
- $A = A_1 \times \ldots \times A_n$, where A_p is a finite set of actions available to player p

- $T : S \times A \times S \to [0, 1]$ is the transition probability function, where $T_{\vec{a}}(s, s')$ is the probability of the process moving to state s' after joint action $\vec{a}$ has been selected in state s
- $R = R^1, \dots R^n$, where $R^p : S \times A \to \mathbb{R}$ is a real-valued payoff function for player p.

At each stage t, the game is in a specific state $s \in S$ and each player $p \in N$ simultaneously chooses an action a_p, such that $\vec{a} = (a_1, \dots, a_n)$ is the joint action. The game stochastically moves into the next state s' according to the transition function $T_{\vec{a}}(s, s')$, and the payoff function $R^p_{\vec{a}}(s, s')$ determines the reward to each player p [Shapley, 1953; Shoham and Leyton-brown, 2009].

2.2 Reinforcement-learning algorithms

Reinforcement-learning problems have been tackled by a variety of approaches. One can categorize algorithms based on the information they require to be available to them [Crandall et al., 2011; Kaelbling et al., 1996]. *Model-free* algorithms learn a policy without learning a model of the environment. In contrast, *model-based* algorithms are given or learn a model of the environment approximating state transition and reward functions, and subsequently use this model to derive a good policy. In order not to convolute the interactive learning process with model building artifacts, this dissertation solely considers model-free algorithms. Since the focus is on a deep understanding of the interactive influences of learning, the scope of this dissertation is restricted to three established general learning algorithms that are not tailored to a specific domain. For a comprehensive overview of reinforcement-learning algorithms the interested reader may consult more comprehensive reviews [Busoniu et al., 2008; Kaelbling et al., 1996].

The concepts in the previous section could be conveniently discussed using a deterministic policy $\pi(s)$ that assigns a single action to each state s. This section introduces algorithms that *gradually* improve the policy, which requires the introduction of a stochastic policy. The stochastic policy x assigns probability $x_i(s)$ to play action i in state s. This rule implies $\forall i, s : 0 \leqslant x_i(s) \leqslant 1$ and $\forall s : \sum_i x_i(s) = 1$. The deterministic policy π can be expressed as a special policy, where $\exists i : x_i(s) = 1$.

2.2.1 Cross learning

One of the most basic reinforcement-learning algorithms is *Cross Learning* [Cross, 1973], which originates from the field of psychology to describe observations of human learning behavior. It can be used to describe learning in multi-armed bandit problems or single-state games. An extension to multiple states using *networks of learning algorithms* is given in Section 4.2.4. At each iteration, the behavior of the agent can be described by the policy $x = (x_1, \dots, x_k)$, which indicates how likely any available action is to be played. The algorithm depends on the reward R_j given in response to action j. However, it is the same no matter if rewards follow a simple distribution (like in multi-armed bandit problems) or whether they are dependent on a joint action or state, hence notation for any further dependencies is omitted. One pure strategy is drawn according to the probabilities and the agent updates its policy x based on the reward R_j received after taking action j:

$$x_i(t+1) \leftarrow x_i(t) + \begin{cases} R_j - R_j x_i(t) & \text{for } i = j \\ -R_j x_i(t) & \text{for all } i \neq j. \end{cases} \tag{2.2}$$

This update maintains a valid policy as long as the rewards are normalized, i.e., as long as $0 \leqslant R_j \leqslant 1$. At each iteration, the probability of the played action j is pushed towards its payoff with the aim of increasing the probability of actions with high expected payoff and decreasing the probability of playing worse strategies. In vector notation,

$$x(t+1) \leftarrow (1 - R_j)\, x(t) + e_j R_j,$$

where e_j is the j^{th} unit vector with all elements zero except for a one at the j^{th} position. The term $1 - R_j$ maintains the probability vector by scaling the previous policy down, such that R_j can be added to the probability of the played strategy. Thus, at each iteration the probability of the selected action is increased unless the payoff is exactly zero. It can be observed that two factors influence the speed of each update: (1) actions with higher payoffs give rise to a larger step in the policy, and (2) actions that are selected more often are reinforced more frequently. The effect of these factors also become apparent in the learning dynamics derived in Section 2.5.1.

Cross learning is closely related to Finite Action-set Learning Automata (FALA) [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}) scheme and a learning step size of 1, and all insights to Cross learning thus equally apply to this type of learning automaton.

2.2.2 Regret minimization

The notion of *Regret Minimization* (RM) forms the basis for another type of reinforcement-learning algorithm. Like Cross learning and learning automata, it has been defined for multi-armed bandit settings or one-state games, and can be extended to multiple states through a network of learning algorithms as discussed in Section 4.2.4. The Polynomial Weights algorithm assumes the best possible payoff is known in hindsight, such that the learner can calculate the loss (or regret) l_j of taking action j rather than the best action in hindsight as $l_j = R^* - R_j$ where R_j is the reward received, and R^* is the optimal reward, i.e., the maximal reward that could have been achieved [Blum and Mansour, 2007]. The learner maintains a set of weights w for its actions, updates the weight of the selected action j according to the perceived loss, and derives a new policy by normalization:

$$\begin{aligned} w_i(t+1) &\leftarrow \begin{cases} w_i(t)\,[1 - \alpha l_i(t)] & \text{for } i = j \\ w_i(t) & \text{for all } i \neq j. \end{cases} \\ x_i(t) &\leftarrow \frac{w_i(t)}{\sum_j w_j(t)} \end{aligned} \tag{2.3}$$

Let $\underline{1} = (1, \dots, 1)$ denote a vector of length k, where k is the number of actions. In vector notation, the weight update reads as follows:

$$w(t+1) \leftarrow w(t)\left(\underline{1} - e_j \alpha l_j(t)\right).$$

Like Cross learning, this algorithm ensures a valid policy as long as the rewards (and thereby losses) are normalized. However, it requires more information, since it needs to know the optimal reward in hindsight. Note that for consistency with the literature, Section 2.5.2 gives the learning dynamics under the assumption that all actions are updated at every step [Klos et al., 2010]. This idealized variant of regret matching requires not only the optimal reward but also a sample reward for all actions not taken. The idealized update in vector notation becomes:

$$w(t+1) \leftarrow w(t)\left(\underline{1} - \alpha l(t)\right),$$

where $l(t) = (l_1(t), \dots, l_k(t))$ is the vector bearing a loss corresponding to each action that could have been taken.

2.2.3 Q-learning

Arguably the most influential reinforcement-learning algorithm is *Q-learning* [Sutton and Barto, 1998; Watkins and Dayan, 1992]. Q-learning is an algorithm that learns good policies for Markov decision processes without having access to the transition and payoff functions. It can also be applied to stochastic games, in which case the reward also depends on actions chosen by other players (see Section 4.2).

Recall that in Markov decision processes, at each time step t the learner selects an action j from its available actions in state s and the process stochastically moves into state s', providing reward $R_j(s, s')$ to the agent in return. **Q-learning** incrementally improves its estimation $Q_j(s, t)$ of the sum of discounted future rewards for taking action j in state s and following a greedy policy thereafter. The action-value is updated according to the following equation, known as the Q-learning update rule, where α denotes the learning rate and γ is the discount factor:

$$Q_j(s, t+1) \leftarrow Q_j(s, t) + \alpha\left(R_j(s, s') + \gamma \max_i Q_i(s', t) - Q_j(s, t)\right). \qquad (2.4)$$

The max operator is used to bootstrap the value of the greedy policy, i.e., the estimate of the best action in the subsequent state is used to update the estimate in this state. Since the executed policy may differ from the policy whose value is estimated, this method is called *off-policy*. In the Q-learning variant **SARSA** (state-action-reward-state-action), the Q-values approximate the value of the executed policy by sampling the next Q-value. SARSA is thus *on-policy*.

$$Q_j(s, t+1) \leftarrow Q_j(s, t) + \alpha\left(R_j(s, s') + \gamma Q_i(s', t) - Q_j(s, t)\right),$$

where action i is drawn according to the policy $x_i(s')$ in the following state s'.

Both Q-learning and SARSA derive the policy from the Q-values. Let $x\left(Q_{\cdot}(\cdot,\cdot),\ldots\right) = (x_1,\ldots,x_k)$ be a function that associates any set of Q-values with a policy, where k is the number of actions, and x_i denotes the probability of selecting action i, such that $x_i \geqslant 0$ and $\sum_{i=1}^{k} x_i = 1$. Various schemes exist to derive the policy, which mainly differ in the way they balance exploration and exploitation. The most prominent examples of such policy-generation schemes are the greedy, ϵ-greedy and the Boltzmann exploration scheme [Sutton and Barto, 1998]. The greedy policy selects the action with the highest Q-value with probability 1. It does not explore at all and is prone to getting stuck in local optima [Sutton and Barto, 1998]. The ϵ-greedy policy chooses the action with the highest Q-value with probability $1-\epsilon$ and a random action with probability ϵ. This policy does explore, but it drastically changes the policy when another action attains the highest Q-value. In contrast, Boltzmann exploration smoothly balances exploration and exploitation by way of a temperature parameter τ. It is defined by the softmax activation function, mapping Q-values to policies:

$$x_i\big(Q(s,t),\tau\big) = \frac{e^{\tau^{-1}Q_i(s,t)}}{\sum_j e^{\tau^{-1}Q_j(s,t)}}. \tag{2.5}$$

The parameter τ lends its interpretation as temperature from the domain of physics. High temperatures lead to stochasticity and random exploration, selecting all actions almost equally likely regardless of their Q-values. In contrast, low temperatures lead to greedy policies with high exploitation of the Q-values, selecting the action with the highest Q-value with probability close to one. Intermediate values prefer actions proportionally to their relative competitiveness. In many applications, the temperature parameter is decreased over time, starting with high exploration and eventually exploiting the knowledge encoded in the Q-values. The policy-generation function ensures a valid policy independent of the reward range, and does not require the reward function to be known or normalized.

Section 2.5.3 presents a simplified model of one-state Q-learning assuming a constant temperature for analytical tractability [Tuyls et al., 2006, 2003]. In addition, derivations assume all actions would be updated at every time step. Chapter 3 critically analyzes these idealized Q-learning dynamics and introduces the variation Frequency Adjusted Q-learning that perfectly adheres to the idealized learning dynamics while only updating one action at a time. The learning dynamics are generalized to time-dependent temperatures in Section 4.1, and extended to multiple states in Section 4.2.

2.3 Reinforcement learning in strategic interactions

In multi-agent systems, several autonomous agents have to react to each other strategically, possibly pursuing conflicting goals. Stochastic games provide a reward signal to the individual learner and thereby make it possible to extend techniques from single-agent learning for multi-agent settings. The following section delineates why learning in multi-agent settings is significantly more complex than single-agent learning. Subsequently, the approach that is followed throughout this dissertation is framed by setting it apart from related multi-agent reinforcement-learning approaches.

2.3.1 Challenges of multi-agent reinforcement learning

In multi-agent settings, the environment is only partially observable for each agent, since each agent introduces its own variables that are hidden from other agents, like its policy and auxiliary internal states. Also, an agent cannot necessarily observe the actions taken by other agents. These properties make it difficult for the agent to distinguish between stochasticity of the environment and the influence of other agents.

Algorithms that have been developed for single-agent learning can be applied to multi-agent settings. However, proofs of convergence in single-agent learning commonly depend on the Markov property and do not hold in the multi-agent case [Kaelbling et al., 1996]. More precisely, if the Markov game is in some state s, the state transition $T_{\vec{a}}(s, s')$ to another state s' depends on the joint action $\vec{a}$. However, the actions taken by other agents are not necessarily observable or may be determined by a new stochastic rule, thus the state transition of the environment depends on more information than is available to the agent, and the environment is not Markovian from the agent's point of view.

Due to the distributed nature of multi-agent systems, centralized learning and control is usually not feasible—no agent has the means or authority to command other agents. Distributed reinforcement learning on the other hand is a much better fit to the demands of many applications, as each agent may learn from experience how to cooperate or to compete [Weiß, 1995]. The next section gives more details on how reinforcement learning deals with the above-mentioned challenges.

2.3.2 Multi-agent reinforcement learning

A variety of algorithms have been specifically devised for multi-agent learning, and they vary largely in their assumptions, especially concerning the observability of states and actions of other agents [Busoniu et al., 2008; Crandall et al., 2011; Panait and Luke, 2005; Shoham et al., 2007]. A survey of cooperative learning in multi-agent systems is available [Panait and Luke, 2005]. It summarizes not only reinforcement-learning techniques but also concepts from evolutionary computation, game theory, complex systems, and robotics. The authors identify two distinctive categories of approaches they name *team learning*, where a single learner seeks a joint solution to multi-agent problems, and *concurrent learning*, using multiple learners simultaneously. Team learning requires the ability of aggregating several independent entities under a joint action space learner. This organization may be a viable option for cooperative settings where agents are benign, but it does not fit the more general setting where agents have individual and possibly conflicting interests. In addition, the essence of *Learning against Learning* is in the interaction between learning processes rather than between agents, and for the sake of analysis each joint action learner represents a single learning process. The scope of this dissertation is restricted to concurrent individual learners, since it is specific enough to elicit the essential interaction between learning processes, and general enough to be applicable in both cooperative and competitive settings.

Besides differentiating cooperative and competitive learning, one can categorize algorithms based on the information available to the individual agents. This information may include observing other agents' state to improve coordination [e HauwereDE HAUWERE et al., 2011], or observing other agents' actions [Hu and Wellman, 2003; Littman, 1994]. Several algorithms have been proven to converge to Nash equilibria in self-play, although proofs are commonly limited to two-player two-action games. They do so under various information requirements [Crandall et al., 2011]. Minimax Q-learning observes the actions of other agents [Littman, 1994]. Nash Q-learning requires to observe other agents' rewards and actions, but nevertheless lacks strong convergence guarantees [Hu and Wellman, 2003]. Friend-or-Foe Q-learning improves upon these guarantees under similarly strong assumptions, showing that it converges to exactly those Q-values that Nash Q-learning ought to converge to [Littman, 2001]. The variation Win-or-Learn-Fast Infinitesimal Gradient Ascent (WoLF-IGA) requires the specification of at least one Nash equilibrium payoff and observes not the reward feedback but the gradient of the reward [Bowling and Veloso, 2002]. Weighted Policy Learning (WPL) does not need the Nash equilibrium payoff but still requires the gradient of the reward [Abdallah and Lesser, 2008]. In contrast to these approaches, Chapter 3 provides a proof of convergence for a new variation of Q-learning named *Frequency Adjusted Q-learning*, which only requires *minimal information*, i.e., the same information that would be available in single-agent learning, namely the agent's own actions and rewards.

Inspired by the PAC framework, performance criteria have been set forward for multi-agent settings, with the aim of *convergence, targeted optimality and safety* [Chakraborty and Stone, 2010; Powers and Shoham, 2004]. This framework requires algorithms to converge to a best response for a set of target opponents, and provide a safety payoff to all other opponents. The target opponents must include self-play, which implies that the algorithm needs to converge to a Nash equilibrium in self-play. Unfortunately, these guarantees are hard to attain, and have so far only been achieved for games that are known by the agent, and where actions of all agents can be observed [Chakraborty and Stone, 2010; Powers and Shoham, 2004].

This dissertation is concerned with the analysis of algorithms that are based on minimal information, although idealized assumptions are sometimes made to make the dynamics tractable for a formal analysis. Learning with *minimal information* means that the agents can only recall their own actions and observe their own rewards, while the actions and rewards of other agents are unobservable [Crandall et al., 2011]. For this purpose, the algorithms presented in Section 2.2 are applied to stochastic games. The restriction to minimal information makes the algorithms applicable in many realistic applications.

2.4 Game theory and strategic interactions

Game theory studies games as formal models of strategic interactions. In contrast to reinforcement learning, which evolves around the agent's process of improving its behavior from experience, classical game theory assumes *rational* players that arrive at their strategy by reasoning. Rationality means that an agent is capable and willing to compute its best possible policy given full or partial knowledge about the game at hand. The next section introduces several key concepts from classical game theory, namely the best response, Nash equilibria and Pareto optimality. Perfect rationality is criticized for not being attained in reality, where resource constraints limit the computability and confounding factors limit adherence to rational behavior. Evolutionary game theory replaces the rationality assumption by concepts from evolutionary biology, such as natural selection and mutation, and is explained in Section 2.4.2. These biological operators are reflected in variations of the replicator dynamics that describe the change of a population over time. The evolutionary dynamics can be analyzed by stability criteria that are related to Nash equilibria of classical game theory. Finally, Section 2.4.3 explains how payoffs can be measured from practical applications to make them available for an evolutionary analysis.

2.4.1 Classical game theory

Classical game theory studies strategic conflicts between intelligent reasoning agents [Gibbons, 1992]. These conflicts are modeled as games, such as normal form games introduced in Section 2.1.3. Recall that a normal form game is defined by the tuple (N, A, R).

- N is the set of n players, with n some finite integer.
- $A = A^1 \times \ldots \times A^n$ is the joint action space, where A^p is the set of actions available to player p,
- and $R = R^1 \times \ldots \times R^n$, where $R^p : A \mapsto \mathbb{R}$ denotes the payoff function of player p, i.e., for any joint action $\vec{a}$, $R^p_{\vec{a}}$ returns the payoff to player p.

The players are assumed to choose their actions simultaneously and independently.

Consider the special case of a one-state two-player game, where the payoffs can be given in a bi-matrix form (R, C) that gives the payoff for the row player in R and the column player in C, as indicated in Figure 2.2 (a).

Policy

As normal form games are stateless, the behavior of each player can be described by a probability vector $x^p = (x^p_1, \ldots, x^p_k)$, that assigns a probability x^p_i to each action i. This probability vector is also called a policy or a mixed strategy.

$$x^p : A^p \to [0, 1] \text{ such that } \sum_{i \in A^p} x^p_i = 1.$$

Let $\vec{x} = (x^1, \ldots, x^n)$ denote the joint policy or mixed strategy profile. Furthermore, let $x^{-p} = (x^1, \ldots, x^{p-1}, x^{p+1}, \ldots, x^n)$ denote the same profile without player p's policy. This notation is useful for the description of best responses and Nash equilibria.

The superscript notation is only used for the general case of n-player games. Throughout this dissertation policies in two-player games will be denoted with x rather than x^1 for the row player, and y rather than x^2 for the column player. For two-action games, such as the examples given in Figure 2.2, the policies can be described by $x = (x_1, 1 - x_1)$ and $y = (y_1, 1 - y_1)$, and the joint policy is fully characterized by the pair (x_1, y_1).

Expected payoff

The expected payoff $v^p\left(x^p|x^{-p}\right)$ for playing policy x^p against the set of opponents' mixed strategies x^{-p} can be computed from the sum over the payoffs of all possible pure strategy profiles $\vec{a} \in A$, multiplied by their probability, where $x^q_{a_q}$ denotes the probability of player q to play action a_q:

$$v^p\left(x^p|x^{-p}\right) = E(R^p|\vec{x}) = \sum_{\vec{a} \in A} R^p_{\vec{a}} \prod_{q \in N} x^q_{a_q}.$$

The expected payoff for playing $x_1 = 1$ (row player, Defect) against $y_1 = \frac{1}{2}$ (column player, mixing both actions equally) in the Prisoners' Dilemma given in Figure 2.2 (b) is $v^1(x|y) = v^1\big((1,0), (\frac{1}{2}, \frac{1}{2})\big) = 3$. The expected payoff in this case for playing $x_1 = 0$ (Cooperate) is $v^1(x|y) = v^1\big((0,1), (\frac{1}{2}, \frac{1}{2})\big) = \frac{3}{2}$.

Best response

The best response is the set of policies that have the maximal possible reward given all other players' policies. Due to the rationality assumption, all players are assumed to pick the best action available to them. A mixed strategy x^p is a best response of player p if there is no other mixed strategy y that would lead to a higher reward for this player, given that all other players' strategies x^{-p} remain the same.

$$\mathrm{BR}(x^{-p}) = x^p \text{ iff } \forall y : v^p\left(x^p|x^{-p}\right) \geqslant v^p\left(y|x^{-p}\right).$$

$$\begin{pmatrix} R_{11}, C_{11} & R_{12}, C_{12} \\ R_{21}, C_{21} & R_{22}, C_{22} \end{pmatrix}$$

(a) general payoff bi-matrix game

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

(b) Prisoners' Dilemma

Figure 2.2: An example of the general payoff bi-matrix game for one-state two-player two-action games, and the payoffs for the Prisoners' Dilemma with actions COOPERATE (C) and DEFECT (D). The Nash equilibrium is marked by an asterisk (*).

The best response of the row player to an opponent playing $y_1 = \frac{1}{2}$ in the Prisoners' Dilemma given in Figure 2.2 (b) is $\mathsf{BR}(y) = (1, 0)$, i.e., Defect. In fact, Defect is a best response to any strategy in the Prisoners' Dilemma; it is a *dominating strategy*.

Nash equilibrium

A core solution concept in game theory is the Nash equilibrium. A Nash equilibrium is a joint policy $\vec{x}^*$ for which no player has an incentive for unilateral deviation, i.e., every strategy x^{*p} is a best response to x^{*-p},

$$x^{*p} = \arg\max_{x^p} v^p(x^p|x^{*-p}).$$

The condition can be expressed in matrix notation for two-player games. Let e_i denote the i^{th} unit vector. (x^*, y^*) comprises a Nash equilibrium iff $\forall i : x^* R y^* \geqslant e_i R y^*$ and $x^* C y^* \geqslant x^* C e_i$. The Prisoners' Dilemma given in Figure 2.2 (b) indicates the Nash equilibrium $x_1 = 1, y_1 = 1$ with an asterisk.

Nash equilibria are the primary concept to derive rational behavior in competitive games. For cooperative games, Pareto optimality is of primary interest.

Pareto optimality

A strategy profile $\vec{x}$ Pareto dominates $\vec{x}'$ if and only if all players obtain at least the same reward and at least one player receives a strictly higher reward when $\vec{x}$ is played.

$$\begin{aligned} &\vec{x} \text{ Pareto dominates } \vec{x}' \\ &\quad \text{iff } \forall p \exists q : v^p(\vec{x}) \geqslant v^p(\vec{x}') \wedge v^q(\vec{x}) > v^q(\vec{x}'). \end{aligned}$$

A strategy profile $\vec{x}$ is Pareto optimal if it is not Pareto dominated.

2.4.2 Evolutionary game theory

Evolutionary game theory takes a rather descriptive perspective, replacing hyperrationality from classical game theory by the concept of natural selection from biology [Smith, 1982]. It studies the population development of individuals belonging to one of several species. The two central concepts of evolutionary game theory are the replicator dynamics and evolutionary stable strategies [Taylor and Jonker, 1978]. The replicator dynamics presented in the next subsection describe the evolutionary change in the population. They are a set of differential equations that are derived from biological operators such as selection, mutation and cross-over. The evolutionary stable strategy describes the resulting asymptotic behavior of this population, and several concepts of stability are presented subsequently. For a detailed discussion of evolutionary game theory, the interested reader may consult one of several survey papers [Hirsch et al., 2004; Hofbauer and Sigmund, 2002].

Replicator dynamics

The replicator dynamics from evolutionary game theory formally define the population change over time. A population comprises a set of individuals, where the species that an individual can belong to relate to pure actions available to a learner. The distribution of the individuals on the different species can be described by a probability vector $x = (x_1, \ldots, x_k)$ that is equivalent to a policy for one player, i.e., x_i indicates the probability of playing action i, or the fraction of the population that belongs to species i. Multi-population models relate one population to each agent's policy.

The evolutionary pressure by natural selection can be modeled by the replicator equations. They assume this population evolves such that successful strategies with higher payoffs than average grow while less successful ones decay. Each species i has a Darwinian fitness f_i, which is related to the payoff function R_i that assigns a reward to a performed action. Since the replicator dynamics describe the change of an infinite population, the Darwinian fitness is related to the expected payoff for a specific action, i.e., $f_i = E[R_i]$. Section 2.4.3 describes how f_i can be computed for practical applications using a heuristic payoff table. While R_i may depend on the population distribution, states or other agents, notation of such dependencies is dropped for the general discussion and only included in specific settings. The general form of the **replicator dynamics** reads as follows, and have been widely studied in the evolutionary game theory literature [Gintis, 2009; Hofbauer and Sigmund, 2002; Weibull, 1996]:

$$\frac{dx_i}{dt} = x_i \left[f_i - \sum_j x_j f_j \right]. \tag{2.6}$$

These dynamics are formally connected to reinforcement learning [Börgers and Sarin, 1997; Tuyls and Parsons, 2007; Tuyls et al., 2006]. This relation is explained in Section 2.5, and extended in Chapter 3.

Consider a one-population model, where each individual encounters a random other individual from the population, and the relative payoff of an individual from species i against an individual of species j is given by R_{ij}. This payoff may be interpreted as the reproductive effect of an encounter, and can be given as a payoff matrix R. Using $f_i = E[R_i] = e_i R x$, where e_i is the i^{th} unit vector, the one-population replicator dynamics can be rewritten as follows denoting the transposed of x with x^T:

$$\frac{dx_i}{dt} = x_i \left[e_i R x^T - x R x^T \right].$$

This one-population model is widely used in Chapter 6 in conjunction with heuristic payoff tables as explained in Section 2.4.3.

To study multiple players that learn concurrently, multi-population models need to be constructed. For ease of exposition, the discussion focuses on only two learning players. Thus, two systems of differential equations are necessary, one for each player. This setup corresponds to asymmetric games, where R and C are the payoff matrices for respectively the row and column player, i.e., $E\left[R_i^1\right] = e_i R y$ and $E\left[R_j^2\right] = x C e_j$, and the available actions of the players belong to two different populations, respectively x and y. This mapping translates into the following coupled replicator equations for the two populations, where e_i denote the i^{th} unit vector:

$$\begin{aligned}\frac{dx_i}{dt} &= x_i\left[e_i R y^T - x R y^T\right]\\ \frac{dy_j}{dt} &= y_j\left[x C e_j^T - x C y^T\right].\end{aligned}$$

The change in the fraction playing action i is proportional to the difference between the expected payoffs $e_i A y$ and $x B e_i$ of action i against the mixing opponent, and the expected payoff xRy and xCy of the mixed strategies x and y against each other. Hence, above average actions get stronger while below average actions decay. The replicator dynamics maintain the probability distribution, thus $\sum_i \frac{dx_i}{dt} = 0$. The examples used in this section are constrained to two actions, which implies $\frac{dx_1}{dt} = -\frac{dx_2}{dt}$ and $\frac{dy_1}{dt} = -\frac{dy_2}{dt}$. The policy space is completely described by the unit square (x_1, y_1), in which the replicator dynamics can be plotted as arrows in the direction of $(\frac{dx_1}{dt}, \frac{dy_1}{dt})$. Using $h = (1, -1)$ and eliminating x_2 and y_2, the equations can be reduced to:

$$\begin{aligned}\frac{dx_1}{dt} &= \alpha x_1(1-x_1)\left[y_1 h R h^T + R_{12} - R_{22}\right]\\ \frac{dy_1}{dt} &= \alpha y_1(1-y_1)\left[x_1 h C h^T + C_{21} - C_{22}\right].\end{aligned}$$

A detailed analysis of this formulation of the replicator dynamics in the context of multi-agent learning dynamics is presented in Section 5.2.

Stability criteria

Evolutionary dynamics can be analyzed for several notions of stability. The most basic concept is the **Fixed Point** (FP) as an equilibrium. Let the dynamical system be defined on a number of probability vectors x^p, which represent the population for each player p. A joint policy $\vec{x} = (x^1, \ldots, x^n)$ is a fixed point if and only if $\forall p, i : \frac{dx_i^p}{dt} = 0$.

Nash equilibria (NE) are a subset of fixed points for the replicator dynamics. On the one hand, all actions that are not played will not be picked up, i.e., $x_i^p = 0 \Rightarrow \frac{dx_i}{dt} = 0$. On the other hand, Nash equilibria only mix between strategies with equal payoffs, thus all actions played with positive probability in a policy belonging to a Nash equilibrium yield payoff equal to the payoff of the Nash equilibrium policy, and $e_i f_i^p = \sum_j x_j^p f_j^p$. Hence, if a joint policy $\vec{x}$ is a Nash equilibrium, then it is also a fixed point of the replicator dynamics.

An equilibrium to which trajectories converge and settle is known as an attractor, while a saddle point is an unstable equilibrium at which trajectories do not settle. Attractors and saddle points are very useful measures of how likely it is that a population converges to a specific equilibrium. Each attractor consumes a certain amount of the strategy space that eventually converges to it. This space is also called the **basin of attraction** of the attractor [Hirsch et al., 2004].

Equilibria are **Asymptotically Stable** (AS) if points that start (infinitesimally) close are pushed back towards the equilibrium. Formally, let $\chi(\vec{x}_0, t)$ denote the trajectory point that is reached from initial joint policy $\vec{x}_0$ by following the dynamics for t units of continuous time, then $\vec{x}$ is asymptotically stable if and only if

$$\exists \epsilon \forall \hat{x} : |\vec{x} - \hat{x}| < \epsilon \leftrightarrow \lim_{t \to \infty} \chi(\hat{x}, t) = \vec{x}.$$

A joint policy is an **Evolutionary Stable Strategy** (ESS) if it is Nash and it cannot be invaded by other mixed strategies, i.e.,

$$\forall p, i, \hat{x} \neq \vec{x} : f_i^p(\vec{x}, \vec{x}) \geqslant f_i^p(\hat{x}, \vec{x}) \wedge f_i^p(\vec{x}, \hat{x}) \geqslant f_i^p(\hat{x}, \hat{x}).$$

Evolutionary stable strategies refine asymptotically stable strategies in the replicator dynamics [Hofbauer and Sigmund, 2002]. Overall, the refinements take the following structure:

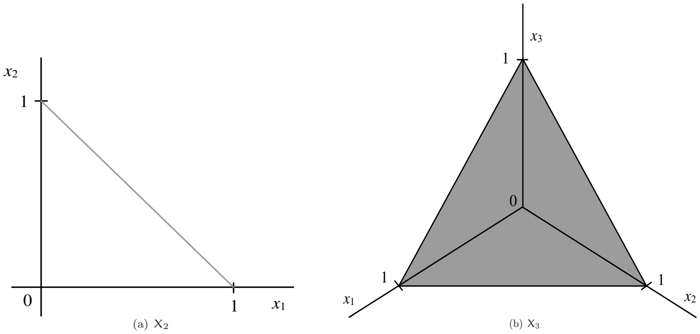
$$\text{ESS} \subseteq \text{AS} \subseteq \text{NE} \subseteq \text{FP}$$

Dynamical systems may also yield cyclic behavior or chaos, in which case trajectories do not settle at fixed points but keep changing forever. Chaos has been observed in two-agent learning dynamics of Q-learning with an epsilon-greedy policy-generation function [Wunder et al., 2010], and cyclic behavior is rather common in the Matching-Pennies game (see Section 5.2.2). The analysis within this dissertation is primarily concerned with investigating the relation between Nash equilibria and asymptotic stability in variations of the replicator dynamics that relate to specific reinforcement-learning algorithms.

Simplex Analysis

The replicator dynamics and other dynamical systems can be visualized in a simplex analysis that facilitates an intuitive grasp of the dynamics. Consider k elements that are randomly chosen with probabilities $x = (x_1, x_2, \ldots, x_k)$, such that $x_1, x_2, \ldots, x_k \geqslant 0$ and $\sum_{i=1}^{k} x_i = 1$. We denote the set of all such probability distributions over k elements as X_k. X_k is a $(k-1)$-dimensional structure and is called a *simplex*. One degree of freedom is lost due to the constraint that the vector be a valid probability vector. Figure 2.3 shows the simplexes X_2 and X_3 for one-population models with two or three actions. Experiments with one-population models mainly use X_3, projected as an equilateral triangle as in Figure 2.3 (b), but dropping the axes and axis labels. As an example, Figure 2.4 shows the game Rock-Paper-Scissors and the simplex plot with arrows that indicate the direction of change $\frac{dx}{dt}$ according to the one-population replicator dynamics.

For multi-population dynamics, the cartesian product of several simplexes is required. As seen in Figure 2.3 (a), the simplex X_2 is a line. In two-player two-action games, the joint policy is completely characterized by the pair (x_1, y_1), the range of which is the unit square $[0, 1]^2$. Throughout the dissertation, experiments in such games are illustrated with dynamics in the unit square, where arrows indicate the change $(\frac{dx_1}{dt}, \frac{dy_1}{dt})$. Figure 2.5 shows the payoff matrix and replicator dynamics for the two-agent game Matching Pennies.



(a) X_2

(b) X_3

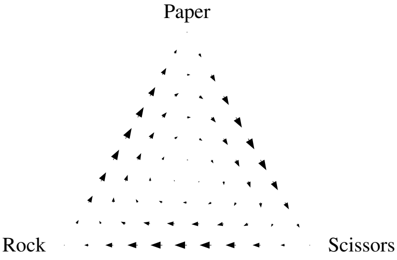
Figure 2.3: The unit simplexes X_2 (a; left) and X_3 (b; right).

2.4.3 Real payoffs and the heuristic payoff table

Real applications are far more complex than benchmark toy domains and a full game representation quickly becomes intractable. The main obstacle is the combinatorial explosion of situations that can arise in complex domains. Instead of analyzing the strategic conflict on the basis of atomic actions, *meta-strategies* or *heuristic strategies* can encapsulate coherent sequences of actions and reactions based on domain knowledge. A player commits to the heuristic strategy before starting the game, and the heuristic fully describes the behavior in the game. The strategic choice is moved to the meta level, and now revolves around which heuristic strategy to choose. This may already yield a much more tractable research problem, but the formulation of heuristic strategies has another advantage: if each action represents a heuristic strategy, then the payoff for that strategy does not depend on which player has chosen it, it rather depends on the composition of strategies it is facing. This setting corresponds to the setting of a *symmetric game*, which is inherent to the use of heuristic strategies.

$$\begin{array}{r|ccc}
 & \mathsf{Rock} & \mathsf{Paper} & \mathsf{Scissors} \\
\mathsf{Rock} & 0 & -1 & 1 \\
\mathsf{Paper} & 1 & 0 & -1 \\
\mathsf{Scissors} & -1 & 1 & 0
\end{array}$$

(a) payoff matrix R

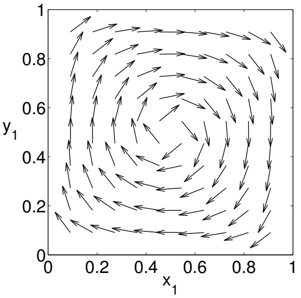


(b) replicator dynamics

Figure 2.4: Payoff matrix and one-population replicator dynamics of the game Rock-Paper-Scissors.

$$\begin{array}{c} \\ \mathit{Head} \\ \mathit{Tails} \end{array} \begin{array}{c} \begin{array}{cc} \mathit{Head} & \mathit{Tails} \end{array} \\ \left(\begin{array}{cc} 2,0 & 0,2 \\ 0,2 & 2,0 \end{array}\right) \end{array}$$

(a) bi-matrix game (R, C)



(b) replicator dynamics

Figure 2.5: Bi-matrix game and two-population replicator dynamics of the Matching Pennies.

Consider a normal form game with n players and k actions. The full representation would require k^n payoff entries and becomes intractably large even for moderate k and n. If however the game is symmetric, the distribution of n players on k pure strategies is a combination with repetition, hence there are $\binom{n+k-1}{n}$ possible meta-strategy compositions. Each of these compositions is a *discrete profile* $\vec{A} = (A_1, \ldots, A_k)$ telling exactly how many players play each strategy. A **heuristic payoff table** captures the payoff information for all possible discrete distributions in a finite population [Walsh et al., 2002].

Suppose we have 3 heuristic strategies and 6 players, this leads to a heuristic payoff table of 28 entries, which is a serious reduction from $3^6 = 729$ entries in the normal form representation. Table 2.1 illustrates what the heuristic payoff table looks like for three strategies A_1, A_2 and A_3. The left-hand side expresses the discrete profile, while the right-hand side gives the payoffs for playing any of the strategies given the discrete profile.

Table 2.1: An example of a heuristic payoff table. The left half of the matrix gives the number of players for each strategy, and the right hand side the average payoff in this distribution. Payoffs of strategies not played are unknown and indicated with a dash.

$$P = \left(\begin{array}{ccc|ccc} A_1 & A_2 & A_3 & U_1 & U_2 & U_3 \\ \hline 6 & 0 & 0 & 0 & - & - \\ & \ldots & & & \ldots & \\ 4 & 0 & 2 & -0.5 & - & 1 \\ & \ldots & & & \ldots & \\ 0 & 0 & 6 & - & - & 0 \end{array}\right)$$

Consider for instance the second row that is given in this table: it shows a profile where 4 players play strategy 1, none of the players play strategy 2 and 2 players play strategy 3. Furthermore, −0.5 is the expected payoff for playing strategy 1 given these set of opponent strategies (i.e., given this discrete profile). When a strategy is not employed by any player, no payoffs are recorded and the resulting expected payoff is then unknown, as indicated with a dash. In zero sum games like poker, discrete profiles where all players play identical strategies yield an expected payoff of 0. In this case, profits and losses are actually divided between the same heuristic strategy deployed by different players, and the average result (for this strategy) is 0.

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

$$P(N_i|x) = \binom{n}{N_{i,1}, N_{i,2}, \ldots, N_{i,k}} \prod_{j=1}^{k} x_j^{N_{i,j}}.$$

If a discrete distribution features zero agents of a certain information type, its payoff cannot be measured and $U_{j,i} = 0$. The expected payoff $f_i(x)$ is computed as the weighted combination of the payoffs given in all rows, compensating for payoff that cannot be measured:

$$f_i(x) = \frac{\sum_j P(N_j|x) U_{j,i}}{1 - (1 - x_i)^k}.$$

This normalized fitness is the basis of the experiments of Chapter 6, which compute the underlying rewards from real world poker plays and simulated double auctions.

2.5 Dynamical systems of multi-agent learning

Evolutionary Game Theory (EGT) has been established as a tool to analyze independent reinforcement learning applied to multi-agent settings [Börgers and Sarin, 1997; Hofbauer and Sigmund, 2002; Tuyls and Parsons, 2007]. Seminal work has shown that Cross learning, a simple policy learner, becomes equivalent to the replicator dynamics when the learning rate is decreased to the infinitesimal limit [Börgers and Sarin, 1997]. The link between learning algorithms and dynamical systems in subsequent work is generally based on the limit of infinitesimal learning rates.

The general procedure for deriving a dynamical system corresponding to the dynamics of a learning algorithms is as follows: The process starts with the difference equation for $\Delta x(t) = x(t+1) - x(t)$; in direct policy search this difference may be taken from the definition of the algorithm while in value iteration (e.g., Q-learning) it needs to be derived from the policy generation function in conjunction with the update rule. Next, suppose that the amount of time that passes between two iterations is 1, and $\delta \Delta x(t)$ makes multiple or fractional updates possible. Using $\delta \to 0$, the continuous time limit of these equations yields a continuous dynamical system. The remainder of this section describes such dynamical systems derived from Cross learning, regret minimization and Q-learning. While $\Delta x(t)$ is a stochastic variable for discrete time steps, the behavior of the dynamical system in the infinitesimal limit is described by the expectation $E[\Delta x(t)]$, and can be interpreted in terms of evolutionary game theory. Each agent's policy relates to a population that describes the distribution over species (actions), and the genetic operators that induce change to the population correspond to the learning rule that updates the agent's policy. This facilitates studying the behavior and convergence properties of learning algorithms by analyzing the corresponding dynamical system.

Tools from dynamical systems make it possible to prove properties of independent reinforcement learning in multi-agent settings, e.g., the average payoff of Infinitesimal Gradient Ascent, a policy gradient learning algorithm, converges to the Nash equilibrium payoff in two-agent two-action matrix games, even though actual policies may cycle [Singh et al., 2000]. This result has been strengthened by introducing the Win-or-Learn-Fast (WoLF) learning speed modulation. The policies of Infinitesimal Gradient Ascent with WoLF learning rates are proven to converge to the Nash equilibrium policies in two-agent two-action games [Bowling and Veloso, 2002]. In contrast to other reinforcement-learning algorithms like Q-learning, Infinitesimal Gradient Ascent assumes that the agents possess a lot of information about the payoff structure[1]. In particular, agents need to compute the gradient of the reward function, which is only possible of the reward function is known. The variations Generalized Infinitesimal Gradient Ascent (GIGA) has been devised to tackle this issue [Bowling, 2005; Zinkevich, 2003], but it is beyond the scope of this dissertation.

The following sections introduce the dynamical systems that have been derived for infinitesimal learning rates in Cross learning and regret minimization. Subsequently, a dynamical system linked to an idealized model of Q-learning is examined in more detail, since it is the basis of the extensions made in Chapter 3. The three algorithms Cross learning, regret minimization and Q-learning are all closely linked to the replicator dynamics from evolutionary game theory. Finally, the related family of dynamical systems of gradient ascent is given in Section 2.5.4. A comparison between all dynamical systems introduced below is drawn in Chapter 5.

2.5.1 Cross learning and the replicator dynamics

Multi-agent learning and evolutionary game theory share a substantial part of their foundation, in that they both deal with the decision-making process of boundedly rational agents 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]. The following paragraphs briefly review this result.

Recall the update rule of Cross learning given in Equation 2.2. Note that the probability x_i of action i is affected both if i is selected and if another action j is selec-

[1] This Infinitesimal Gradient Ascent family is not to be confused with REINFORCE learning automata that estimate the gradient from samples [Williams, 1992], and are thus closer to independent reinforcement learning.

ted. Let R_i or R_j be the reward received for taking action i or j respectively, and let $f_i = E[R_i]$ denote the expectation of the reward. Recall that the policy change $\Delta x_i(t) = x_i(t+1) - x_i(t)$ is time dependent. In expectation, Equation 2.2 induces the following update, where the reference to time is dropped for readability:

$$\begin{aligned} E[\Delta x_i] &= \overbrace{x_i[f_i - f_i x_i]}^{\text{update to this action}} + \overbrace{\sum_{j \neq i} x_j[-f_j x_i]}^{\text{update to other actions}} \\ &= x_i\left[f_i - \textstyle\sum_j x_j f_j\right]. \end{aligned}$$

Let the discrete algorithm assume 1 time unit between updates, then the continuous limit of this process can be taken as $x_i(t+\delta) = x_i + \delta\Delta x_i$, with $\lim \delta \to 0$. This transformation yields a continuous system, which can be expressed with the partial differential equation. The equation is $\frac{dx_i}{dt} = x_i\left[f_i - \sum_j x_j f_j\right]$, which is equivalent to the replicator dynamics of Equation 2.6. For multi-population dynamics, the policy of each player p evolves according to the replicator dynamics for $\frac{x_i^p}{dt}$, where f_i^p depends on the joint policy $\vec{x}$.

The convergence behavior of Cross learning, being a simple learning automaton, has also been classified in terms of Nash equilibria from game theory. In self-play, pure Nash equilibria are found to be stable while mixed Nash equilibria are unstable [Thathachar and Sastry, 2003]. Equivalent results can be derived from the replicator dynamics as a model of the learning process [Hofbauer and Sigmund, 2002].

2.5.2 Learning dynamics of regret minimization

The evolutionary framework has also been extended to the Polynomial Weights algorithm, which as described in Section 2.2.2 implements *Regret Minimization* [Blum and Mansour, 2007]. Despite the great difference in update rule and policy generation (see Eq. 2.3), the infinitesimal limit has been linked to a dynamical system that is quite similar to the dynamics of Cross learning [Klos et al., 2010].

$$\frac{dx_i}{dt} = \frac{\alpha x_i\left[f_i - \sum_j x_j f_j\right]}{1 - \alpha\left[\max_k f_k - \sum_j x_j f_j\right]}.$$

The numerator is equivalent to the replicator dynamics, and thus to Cross learning. The denominator can be interpreted as a learning-rate modulation dependent on the best action's update. For two-player games, the payoffs can be expressed by the bi-matrix game (R, C), and $f_i = e_i R y$ for the first player:

$$\frac{dx_i}{dt} = \frac{\alpha x_i\left[e_i R y^T - x R y^T\right]}{1 - \alpha\left[\max_k e_k R y^T - x R y^T\right]}.$$

The dynamics for the second player are analogous for $\frac{dy_j}{dt}$ and are omitted here.

2.5.3 An idealized model of Q-learning

The infinitesimal limit of Cross learning is equivalent to the replicator dynamics (see Section 2.5.1), and its application to Q-learning reveals very similar dynamics [Tuyls et al., 2003]. However, a simplifying assumption was made to derive an idealized model of Q-learning: Suppose that Q-learning would be updating all actions at each iteration, the dynamics then give rise to the following system of differential equations:

$$\frac{dx_i}{dt} = \underbrace{\tau^{-1} x_i \alpha \left[f_i - \sum_j x_j f_j \right]}_{\text{replicator dynamics}} + \underbrace{x_i \alpha \left(-\log x_i + \sum_k x_k \log x_k \right)}_{\text{exploration terms}}. \tag{2.7}$$

The striking part of this result was that the equations contain a part equal to replicator dynamics, identified to represent natural selection [Weibull, 1996], and additional terms that relate to entropy, and can be considered a model of mutation. Relating entropy and mutation is not new. It is well known [Schneider, 2000; Stauffer, 1999] that mutation increases entropy. The concepts are similar to thermodynamics in the following sense: the selection mechanism is analogous to *energy* and mutation to *entropy* [Stauffer, 1999]. Hence generally speaking, mutations tend to increase entropy. Exploration from reinforcement learning then naturally maps to the mutation concept, as both concepts take care of providing variety. Analogously, selection maps to the greedy concept of exploitation in reinforcement learning. The replicator dynamics encoding selection are scaled inversely proportional to the exploration parameter τ of the Q-learning algorithm. This argument implies that exploration is dominant for large τ, and exploitation is dominant for small τ. Due to the infinitesimal limit, the magnitude of the dynamical system does not change the convergence behavior defined by the direction and proportionality of the force field. For a detailed discussion in terms of selection and mutation operators, the interested reader may consult the references [Tuyls et al., 2006, 2003].

It is known from game theoretic studies that human players do not purely select their actions greedily [Gintis, 2009]. Once in a while they also randomly explore their alternative actions. This finding closely resembles the theory of reinforcement learning where players have to make a trade off between exploration and exploitation [Sutton and Barto, 1998]. In Chapter 6, the idealized model of Q-learning is used as a model of learning with exploration.

With this dynamical model, it is possible to get insight into the learning process, its traces, basins of attraction, and stability of equilibria by just examining the coupled system of replicator equations and plotting its force and directional fields. The learning dynamics for two-player stateless games can be described in matrix notation:

$$\frac{dx_i}{dt} = x_i \alpha \left(\tau^{-1} \left[e_i R y^T - x R y^T \right] - \log x_i + \sum_k x_k \log x_k \right)$$

$$\frac{dy_j}{dt} = y_j \alpha \left(\tau^{-1} \left[x C e_j^T - x C y^T \right] - \log y_j + \sum_l y_l \log y_l \right)$$

with x, y the policies, α the learning rate, τ temperature parameter, R, C the payoff matrices, and e_i the i^{th} unit vector.

Börgers et al. observed that the actual learning traces of Cross learning may deviate from the predicted behavior [Börgers and Sarin, 1997]. Similarly, it can be observed that the behavior of the Q-learning process does not always match the derived Q-learning dynamics. While the correspondence between algorithm and model improves under smaller learning rates in Cross learning, these deviations are systematic and non-negligible for Q-learning. Chapter 3 analyzes why it is the case and presents a variation of Q-learning that perfectly matches the dynamical system.

2.5.4 Dynamical systems of gradient ascent

Gradient ascent (or decent) is a well known and capable optimization technique in the field of machine learning [Sutton and Barto, 1998]. Given a well-defined differentiable objective function, the learning process follows the direction of its gradient in order to find a local optimum. This concept has been adapted for multi-agent learning by improving the learning agents' policies along the gradient of their payoff function. This approach assumes that the payoff function, or more precisely the gradient of the expected payoff, is known to the learners.

One algorithm that implements gradient ascent is **Infinitesimal Gradient Ascent** (IGA), in which a learner updates its policy by taking infinitesimal steps in the direction of the gradient of its expected payoff [Singh et al., 2000]. It has been proven that in two-player two-action games, the joint policy of IGA in self-play 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.

The learning algorithm for repeated (single-state) games is defined as follows: A learner's policy $x(t) = \{x_1, x_2, \ldots, x_k\}$ denotes a probability distribution over its k possible actions at time t, where x_i is the probability of selecting action i, i.e., $\forall i : 0 \leqslant x_i \leqslant 1$, and $\sum_i x_i = 1$. Take $V(x) : \mathbb{R}^n \to \mathbb{R}$ to be the value function that maps a policy to its expected payoff. The policy update rule for IGA can now be defined as

$$\begin{aligned} \Delta x_i(t) &\leftarrow \alpha \frac{\partial V(x(t))}{\partial x_i(t)} \\ x_i(t+1) &\leftarrow \mathsf{projection}(x_i(t) + \Delta x_i(t)) \end{aligned} \tag{2.8}$$

where α denotes the learning step size. The intended change $\Delta x(t)$ may take x outside of the valid policy space, in which case it is projected back to the nearest valid policy by the $\mathsf{projection}$ function.

Win or Learn Fast (WoLF) is a variation on IGA that uses a variable learning rate [Bowling and Veloso, 2002]. 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. The modified learning rule of IGA-WoLF is

$$\begin{aligned}\Delta x_i(t) &\leftarrow \frac{\partial V(x(t))}{\partial x_i(t)} \begin{cases} \alpha_{min} & \text{if } V(x(t)) > V(x^*) \\ \alpha_{max} & \text{otherwise} \end{cases} \\ x_i(t+1) &\leftarrow \mathtt{projection}(x_i(t) + \Delta x_i(t))\end{aligned} \tag{2.9}$$

where x^* is a policy belonging to an arbitrary Nash equilibrium. The presence of x^* in the algorithm means that WoLF needs not only to know the value function but also at least one strategy that is part of a Nash equilibrium.

The **Weighted Policy Learner** (WPL) is a second variation of IGA that also modulates the learning rate, but in contrast to WoLF-IGA it does not require knowledge of Nash equilibria [Abdallah and Lesser, 2008]. The update rule of WPL is defined as

$$\begin{aligned}\Delta x_i(t) &\leftarrow \alpha \frac{\partial V(x(t))}{\partial x_i(t)} \begin{cases} x_i(t) & \text{if } \frac{\partial V(x(t))}{\partial x_i(t)} < 0 \\ 1 - x_i(t) & \text{otherwise} \end{cases} \\ x_i(t+1) &\leftarrow \mathtt{projection}(x_i(t) + \Delta x_i(t))\end{aligned} \tag{2.10}$$

where the update is weighted either by x_i or by $1 - x_i$ depending on the sign of the gradient. The $\mathtt{projection}$ function is slightly different from the one used in IGA, in that the policy is projected to the closest valid policy that lies at distance $\epsilon > 0$ to the policy space boundary, i.e., $\forall t, i : \epsilon \leqslant x_i(t) \leqslant 1 - \epsilon$.

2.6 Summary and limitations of the state of the art

This chapter has given the most relevant concepts for multi-agent learning from the domains of reinforcement learning, game theory and dynamical systems. Throughout the remainder of the dissertation, these concepts are interrelated to provide a comprehensive grasp of learning algorithms in strategic interactions.

Strategic interactions are formalized in repeated normal form games (single-state) or stochastic games (multi-state). Reinforcement-learning algorithms are situated in these games and iteratively improve their policy based on the experienced payoff signal while balancing a good performance (exploitation) with gaining experience that improves the knowledge (exploration). The prerequisites for multi-agent learning algorithms are diverse, and some algorithms require observing other agents' actions and rewards.

The algorithms can either be studied by testing them empirically, or the convergence behavior can be described analytically. For the latter purpose, learning algorithms are linked to a set of differential equations that describes a dynamical system of their learning dynamics given infinitesimal learning rates. For the three discussed learning strategies, Cross learning, regret minimization and Q-learning, the learning dynamics are closely related to the replicator dynamics from evolutionary game theory. However, the learning dynamics of Q-learning have been derived under the simplifying assumption that Q-learning would update all actions at every iteration, because without this assumption the learning dynamics cannot be expressed in the policy space and are thus not tractable in the evolutionary framework.

The learning dynamics can be proven to cycle or to converge to certain joint policies. If every player converges to a best reply, the joint policy constitutes a Nash equilibrium of the normal form game, and thereby links the learning behavior to classical game theory. The strongest guarantees have been derived for algorithms that need a lot of information, e.g., convergence to Nash equilibria is guaranteed for Minimax Q-learning and Friend-or-Foe Q-learning which require observing other agents (see Section 2.3). The same convergence has been attained for variations of Infinitesimal Gradient Ascent, which require the gradient of the reward to be known. Chapter 3 improves the state of the art in two ways: (1) it introduces the variant Frequency Adjusted Q-learning (FAQ-learning) that corresponds to the idealized dynamical model of Q-learning, and (2) it provides a proof of convergence to Nash equilibria for this new variant that only requires minimal information. The idealized model of Q-learning is based on a constant exploration rate and only available for single-state games, although many applications are best modeled as multi-state environments and single-agent learning theory reveals that probability of convergence to global optima is increased by using an initially high, then decreasing exploration. Chapter 4 derives the more general model of FAQ-learning under time-dependent exploration rates, and extends the model to stochastic games. The gradient ascent dynamics are suggestively similar to the replicator dynamics although they start from different information requirements. The relationship between the replicator dynamics and gradient ascent dynamics is investigated in detail in Chapter 5.

3 Frequency Adjusted Q-learning

Various derivatives of Q-learning play a prominent role in single- and multi-agent reinforcement learning [Busoniu et al., 2008; Crandall et al., 2011; Shoham et al., 2007; Sutton and Barto, 1998; Watkins and Dayan, 1992]. The process of Q-learning starts from some initialization of Q-values, which may encode optimistic, neutral or pessimistic priors. While these initial parameter values may be arbitrary in single-agent settings [Watkins and Dayan, 1992], they may become crucial in multi-agent learning [Crandall et al., 2011]. The analysis below reveals that this dependency on initial values may lead to irrational policy trajectories, i.e., the probability of a dominating action is not monotonically increased (see Figure 3.1). In contrast, the idealized

$$\begin{array}{ccc}
 & \begin{array}{cc} \mathrm{D} & \mathrm{C} \end{array} & \\
\begin{array}{c} \mathrm{D} \\ \mathrm{C} \end{array} & \begin{pmatrix} 1,1 & 5,0 \\ 0,5 & 3,3 \end{pmatrix} & \begin{array}{c} x \\ 1-x \end{array} \\
 & \begin{array}{cc} y & 1-y \end{array} &
\end{array}$$

Prisoner's Dilemma

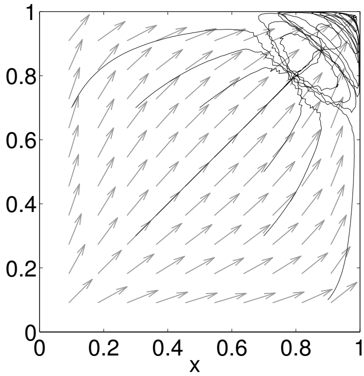


Figure 3.1: Trajectories of Q-learning with a neutral prior (solid lines) and dynamics of its idealized model (arrows) in the Prisoner's Dilemma.

evolutionary model presented in Section 2.5.3 yields rational policy progression. This discrepancy motivates modifying the algorithm to match the idealized model.

In this chapter, the discrepancy between Q-learning and its idealized model is examined further. Based on analytical insights, the Frequency Adjusted Q-learning (FAQ-learning) algorithm is proposed. This variation of Q-learning inherits the behavior of the idealized model for an arbitrarily large part of the policy space. In addition to the theoretical discussion, experiments in the three classes of two-agent two-action games illustrate the superiority of FAQ-learning. Finally, a proof of convergence in two-player two-action games contributes to the theoretical foundation of the algorithm, and more generally to the analytical framework of multi-agent reinforcement learning. This chapter is based on prior publications [Kaisers and Tuyls, 2010, 2011].

3.1 Discrepancy between Q-learning and its idealized model

Q-learning updates the state-action value estimates whenever a specific action is selected. As a result, each estimate is updated at its own frequency, and estimates of actions that are selected more often are updated faster. In contrast, the idealized model assumes all estimates to be updated equally fast. In essence, the newly proposed variation needs to compensate for the difference in frequencies by modulating the learning step size for each action separately. As a result, initialization dependencies are removed and convergence progresses through more rational policy trajectories, i.e., in expectation never moving away from dominant actions. It has been shown that modulating the learning rate can improve learning performance, e.g., Bowling et al. [Bowling and Veloso, 2002] have modulated the learning rate anti-proportional to the success of the current strategy. The approach presented here is different in that it considers the learning rate of each action separately, compensating for the fact that an action that is selected more often receives more updates and thereby has its value estimates updated more quickly.

The standard Q-learning algorithm only updates the Q-value associated with the selected action. For the simplicity of illustration, consider the single-state Q-learning algorithm, where notation for state-dependency is dropped, i.e., after action a is selected the respective Q-value is changed according to:

$$Q_a(t+1) \leftarrow Q_a(t) + \alpha \left(r_a(t) + \gamma \max_j Q_j(t) - Q_a(t) \right).$$

This can be rewritten to describe the change of the Q-value associated with an arbitrary action i:

$$\begin{aligned} \Delta Q_i(t) &= Q_i(t+1) - Q_i(t) \\ &= \begin{cases} \alpha \left(r_i(t) + \gamma \max_j Q_j(t) - Q_i(t) \right) \text{ if i=a} \\ \hfill 0 \text{ otherwise.} \end{cases} \end{aligned}$$

The policy x determines the frequency with which each Q-value is updated and it influences the expected Q-value change. The expected reward $E[r_i(t)]$ also depends on the environment and the other agents. The resulting expected Q-value change incurred by the Q-learning update rule is thus:

$$E[\Delta Q_i(t)] = x_i(t) \cdot \alpha \Big(E[r_i(t)] + \gamma \max_j Q_j(t) - Q_i(t) \Big).$$

Other authors [Babes et al., 2009; Gomes and Kowalczyk, 2009; Wunder et al., 2010] independently arrived at the same expected change of Q-values. However, these sources explicitly consider ϵ-greedy as the policy generation function, which maps Q-values to a few discrete policies and thus does not allow the policy space of the process to be described in a self-consistent way.

The dynamical system associated with Q-learning is based on the continuous time limit of the learning process. It is inspired by prior work [Börgers and Sarin, 1997], which describes a policy learner with infinitesimal time steps and shows that the process of multi-agent *Cross-learning* converges to the replicator dynamics in the continuous time limit. In the learning algorithm, updates proceed in discrete iterations of $\Delta t = 1$,

$$E[Q_i(t+1) - Q_i(t)] = 1 \cdot x_i(t) \cdot \alpha \Big(E[r_i(t)] + \gamma \max_j Q_j(t) - Q_i(t) \Big).$$

If this update is incurred twice, $\Delta t = 2$. By similar reasoning one can generalize this to fractional updates. The continuous time limit can be constructed by changing the basis for time to δ, representing an arbitrary multiple or fractional update time interval, and then taking the limit of δ to zero:

$$E[Q_i(t+\delta) - Q_i(t)] = \delta \cdot x_i(t) \cdot \alpha \Big(E[r_i(t)] + \gamma \max_j Q_j(t) - Q_i(t) \Big).$$

Now, taking the continuous time limit $\delta \to 0$, and using the dot notation for differentiation with respect to time, this equation becomes:

$$E\left[\dot{Q}_i\right] = x_i(t) \cdot \alpha \Big(E[r_i(t)] + \gamma \max_j Q_j(t) - Q_i(t) \Big).$$

Consider that $\dot{Q}_i$ is continuous and can be treated as constant over an infinitesimally small area with diameter ϵ, and in the infinitesimal time limit of $\alpha \to 0$ an infinite number of updates is perceived between two distinct policies. As a consequence, the expected change equals the actual change in that limit, i.e., $E\left[\dot{Q}_i\right] = \dot{Q}_i$. Formally, $\forall \epsilon > 0$, however small, $\exists \delta > 0 : \epsilon = k\delta$, with $k \to \infty$ and $E\left[\frac{dQ_i}{\epsilon}\right] = E\left[\frac{dQ_i}{k\delta}\right] = \frac{1}{k} E\left[\frac{dQ_i}{\delta}\right]$. According to the law of large numbers, the mean approaches the expected value for large k. Hence, $\frac{1}{k} E\left[\frac{dQ_i}{\delta}\right] = \frac{dQ_i}{k\delta} = \frac{dQ_i}{\epsilon}$.

$$\dot{Q}_i = x_i(t) \cdot \alpha \Big(E[r_i(t)] + \gamma \max_j Q_j(t) - Q_i(t) \Big).$$

Finally, the difference between the Q-learning dynamics derived here and the idealized model can be observed by juxtaposition. The idealized model [Tuyls et al., 2006, 2003] starts from the following dynamics, assuming simultaneous updates of all actions:

$$\dot{Q}_i = \alpha \left(E\left[r_i(t)\right] + \gamma \max_j Q_j(t) - Q_i(t) \right).$$

This equation differs from the original $\dot{Q}_i$ by a factor of $x_i(t)$, and explains observed anomalies (deviations from the idealized model) and initialization dependencies of the update rule. This discrepancy can be resolved in two ways: one, describing the dynamics of Q-learning more precisely by incorporating the factor $x_i(t)$, or two, adapting the Q-learning update rule to fit the model.

The idealized evolutionary game theoretic model describes *more rational* policy trajectories than Q-learning actually exhibits (see Figure 3.1). For example, if two actions are over-estimated by the current Q-values and a dominant action receives more updates due to being selected more often, the dominant action will lose its over-estimation more quickly and Q-learning may policy-wise move away from this dominant action. Such behavior is undesirable because the problem of over- and under-estimation is prevalent in the application of the algorithm. The Q-values need some initialization that must not be based on knowledge of the rewards. An inappropriate initialization leads to errors in the estimates. Analyses of single-agent learning may overcome these dependencies by focusing on the limit of infinite time or by sufficient initial exploration [Watkins and Dayan, 1992], but the amount of exploration that suffices may differ from case to case and if underestimated, i.e., if exploration is decreased prematurely, the same problems of wrong estimates re-occur. Furthermore, the performance in interaction with other learning agents may greatly depend on the priors and initial behavior [Crandall et al., 2011]. Another drawback of moving away from dominant actions is the decrease of expected reward for a period of time, which may in some applications be worse than an almost monotonically ascending expected reward with a slightly lower accumulated payoff. For example, humans commonly prefer monotonically increasing income over temporarily decreasing income, even if the cumulative reward is lower [Ariely, 2009]. For these reasons, rather than striving for a more precise description of Q-learning, I propose an alternative update rule for Q-learning in the next section, i.e., Frequency Adjusted Q- (FAQ-) learning that perfectly matches the dynamical system.

3.2 Implementing the idealized model of Q-learning

This section introduces Frequency Adjusted Q-learning (FAQ-learning), which inherits the more desirable game theoretic behavior of the evolutionary game theory model that was derived from idealized assumptions about Q-learning. In particular, the update rule is adapted to compensate for the frequency term $x_i(t)$ in the expected Q-value change. A comparison between Q-learning, FAQ-learning and the idealized dynamics illustrates the merits of the newly proposed FAQ-learning algorithm in multi-agent settings.

3.2.1 The algorithm Frequency Adjusted Q-learning

FAQ-learning is equivalent to Q-learning, except for the update rule for which it uses the following adapted version:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i(t)}\alpha\left(r_i(t) + \gamma \max_j Q_j(t) - Q_i(t)\right). \tag{3.1}$$

Using the same reasoning as in the previous section, the continuous time limit of this process converges to the following equation:

$$\dot{Q}_i = \alpha\bigg(\mathsf{E}\left[r_i(t)\right] + \gamma \max_j Q_j(t) - Q_i(t)\bigg). \tag{3.2}$$

As a result, FAQ-learning yields precisely the dynamics of the idealized model [Tuyls et al., 2006, 2003], while classical Q-learning differs by a factor of $x_i(t)$. In contrast to previous work, FAQ-learning does not need idealizing assumptions, since it balances frequencies by modulating the learning rate for each action individually. The experiments in the next section show how this difference translates to an exact match of FAQ-learning and the evolutionary game theoretic model, while anomalies and differences between Q-learning and its idealized model can be observed.

The update rule requires an adaptation to be applied as an algorithm in practical domains or for numeric analysis, since the algorithm written in Equation 3.1 is only valid in the infinitesimal limit of α, otherwise $\frac{\alpha}{x_i(t)}$ may become larger than 1. This situation would allow the Q-values to escape the convex hull of experienced rewards. That in turn is unreasonable for learning. In fact, a maximal learning step should be very small to yield reasonable convergence behavior, i.e., $\frac{\alpha}{x_i(t)} << 1$. Consequently, this formal version of FAQ-learning cannot be applied numerically yet. I propose the following generalized and practical model of FAQ-learning with a new model parameter $\beta \in [0, 1]$:

$$Q_i(t+1) \leftarrow Q_i(t) + \min\left(\frac{\beta}{x_i(t)}, 1\right)\alpha\bigg(r_i(t) + \gamma \max_j Q_j(t) - Q_i(t)\bigg).$$

Next, inspect the properties of this update rule, considering that the behavior changes at $\frac{\beta}{x_i(t)} = 1$, which is at $x_i(t) = \beta$. For notational convenience, the time dependency is dropped from $x_i(t)$, $Q_i(t)$, and $r_i(t)$ in the following equation:

$$\begin{aligned} x_i \geqslant \beta : \mathsf{E}\left[\Delta Q_i\right] &= \frac{\beta}{x_i}\alpha\bigg(\mathsf{E}\left[r_i\right] + \gamma \max_j Q_j - Q_i\bigg) \\ x_i < \beta : \mathsf{E}\left[\Delta Q_i\right] &= \quad\alpha\bigg(\mathsf{E}\left[r_i\right] + \gamma \max_j Q_j - Q_i\bigg). \end{aligned}$$

If $\beta = 1$, this model degenerates to classical Q-learning. If $0 \leqslant \beta < 1$, the limit of $\alpha \to 0$ makes this model equivalent to formal FAQ-learning with a learning rate of $\alpha\beta$, i.e., the behavior converges to the derived replicator dynamics [Tuyls et al., 2006, 2003]. Numerical simulation needs to choose finitely small α. In that case, the dynamics for $x_i \geqslant \beta$ are equivalent to FAQ-learning with learning rate $\alpha\beta$, while the dynamics for $x_i < \beta$ equal those of classical Q-learning with learning rate α. Hence, the maximal learning step is defined by α and needs to be reasonably small, while the size of the subspace that behaves like FAQ-learning is controlled by β. For both parameters, smaller values are more desirable regarding the path of convergence, but lead to an increase in the required number of iterations. By choosing β arbitrarily small, the learner behaves according to the evolutionary model for an arbitrarily large part of the policy space. The examples given below empirically evaluate FAQ-learning with $\beta = \alpha$ to obtain a smooth convergence to the true Q-values, while maintaining the preferred update behavior for a large part of the policy space.

3.2.2 Experiments and results

This section compares Q-learning and FAQ-learning trajectories to the idealized model. For the sake of clarity, the empirical evaluation is restricted to two-player two-action normal form games. This type of game can be characterized as a payoff bi-matrix game (A, B), where for any joint action (i, j) the payoff to player one and two are given by A_{ij} and B_{ij}, respectively. Figure 3.2 shows three representative classes of two-action two-player games [Tuyls et al., 2006, 2003]: the Prisoners' Dilemma (PD), the Battle of Sexes (BoS), and Matching Pennies (MP). They represent the classes of games with one pure Nash Equilibrium (PD), with one mixed and two pure Nash Equilibria (BoS), and with one mixed Nash Equilibrium (MP).

For Boltzmann action selection, policies do not uniquely identify the Q-values they are generated from. Translation of all Q-values by an equal amount does not alter the policy, which is solely dependent on the difference between the Q-values. For example, the Q-value pair $(0, 1)$ generates the same policy as $(1, 2)$. The replicator dynamics describe the policy change as a function of the policy, while the learning update rule incurs a policy change dependent on the policy and the Q-values. To compare Q-learning and FAQ-learning to the evolutionary dynamics, learning trajectories showing the update rule's effect are given for several translations of initial Q-values. In particular, the initial Q-values are centered around the minimum, mean or maximum possible Q-value, given the game's reward space. As such, they encode pessimistic, neutral or optimistic priors. Since Q-values estimate the discounted sum of future rewards, their range relates to the reward according to the following equation for the minimum $Q_{min} = \sum_{i=0}^{\infty} \gamma^i r_{min} = \frac{1}{1-\gamma} r_{min}$, and similarly for the maximum value.

$$\begin{array}{c} \\ D \\ C \end{array} \begin{array}{c} \begin{array}{cc} D & C \end{array} \\ \left(\begin{array}{cc} 1,1 & 5,0 \\ 0,5 & 3,3 \end{array}\right) \end{array} \qquad \begin{array}{c} \\ B \\ S \end{array} \begin{array}{c} \begin{array}{cc} B & S \end{array} \\ \left(\begin{array}{cc} 2,1 & 0,0 \\ 0,0 & 1,2 \end{array}\right) \end{array} \qquad \begin{array}{c} \\ H \\ T \end{array} \begin{array}{c} \begin{array}{cc} H & T \end{array} \\ \left(\begin{array}{cc} 2,0 & 0,2 \\ 0,2 & 2,0 \end{array}\right) \end{array}$$

Figure 3.2: Reward matrices for Prisoners' Dilemma (left, *Defect* or *Cooperate*), Battle of Sexes (right, *Bach* or *Stravinski*) and Matching Pennies (bottom, *Head* or *Tail*).

The neutral initialization is centered between the minimum and maximum value. If $\gamma = 0$, this gives rise to $\{0, 2\frac{1}{2}, 5\}$ for the Prisoners' Dilemma, and $\{0, 1, 2\}$ for Battle of Sexes and for Matching Pennies; with $\gamma = 0.9$ the range increases to the tenfold. Figures 3.4 and 3.3 show trajectories obtained from running the learners with $\gamma = 0.9$, $\alpha = 10^{-6}$ for Q-learning, and $\alpha = \beta = 10^{-3}$ for FAQ-learning, with a fixed temperature $\tau = 0.1$. The trajectories yield 200 thousand iterations in all but the neutral and optimistic Q-learning in the Prisoners' Dilemma, which use 500 thousand iterations.

While classical Q-learning shows significantly different learning behavior depending on the initialization, FAQ-learning merely increases the noise for higher values in the initialization. The noise is caused by larger learning steps, as the Q-value change includes a term $-\alpha Q_i(t)$, which is clearly proportional to the magnitude of the Q-values. Nonetheless, the expected direction of change remains unaffected in FAQ-learning.

In comparison to the evolutionary prediction, the FAQ-learning trajectories always follow the predicted expected change, while Q-learning trajectories deviate from it depending on the initialization. The behavior of Q-learning and FAQ-learning are most similar to each other for the mean reward initialization. However, tweaking the initialization does not remove but only reduces the deviations, and knowing the exact reward space violates the assumption of many applications. In addition, the Prisoners' Dilemma shows qualitatively significant differences even for the mean initialization. In sum, the behavior of FAQ-learning is consistent across initializations, which Q-learning is not, and exhibits rational policy improvements in line with the idealized model.

Matching Pennies

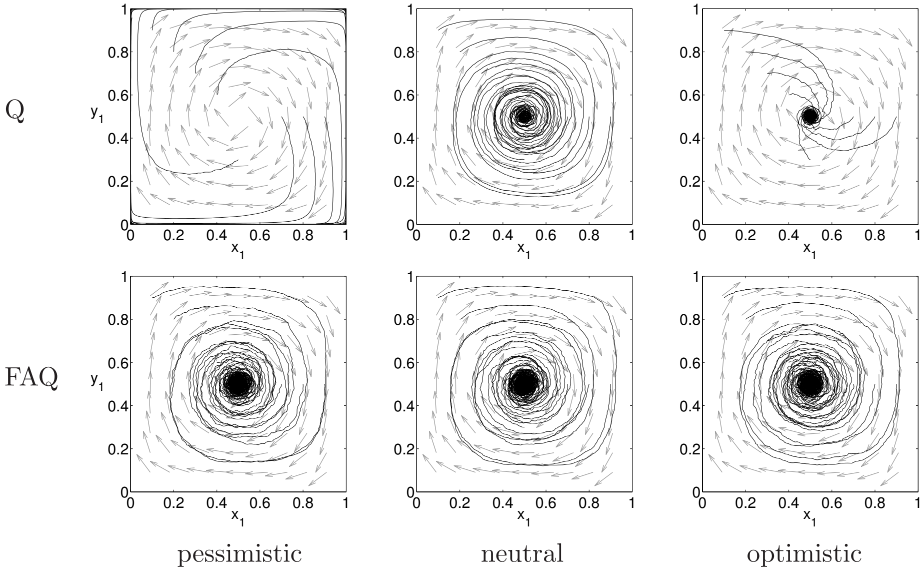


Figure 3.3: Comparison of Q-learning to FAQ-learning with various Q-value initializations in the Matching Pennies.

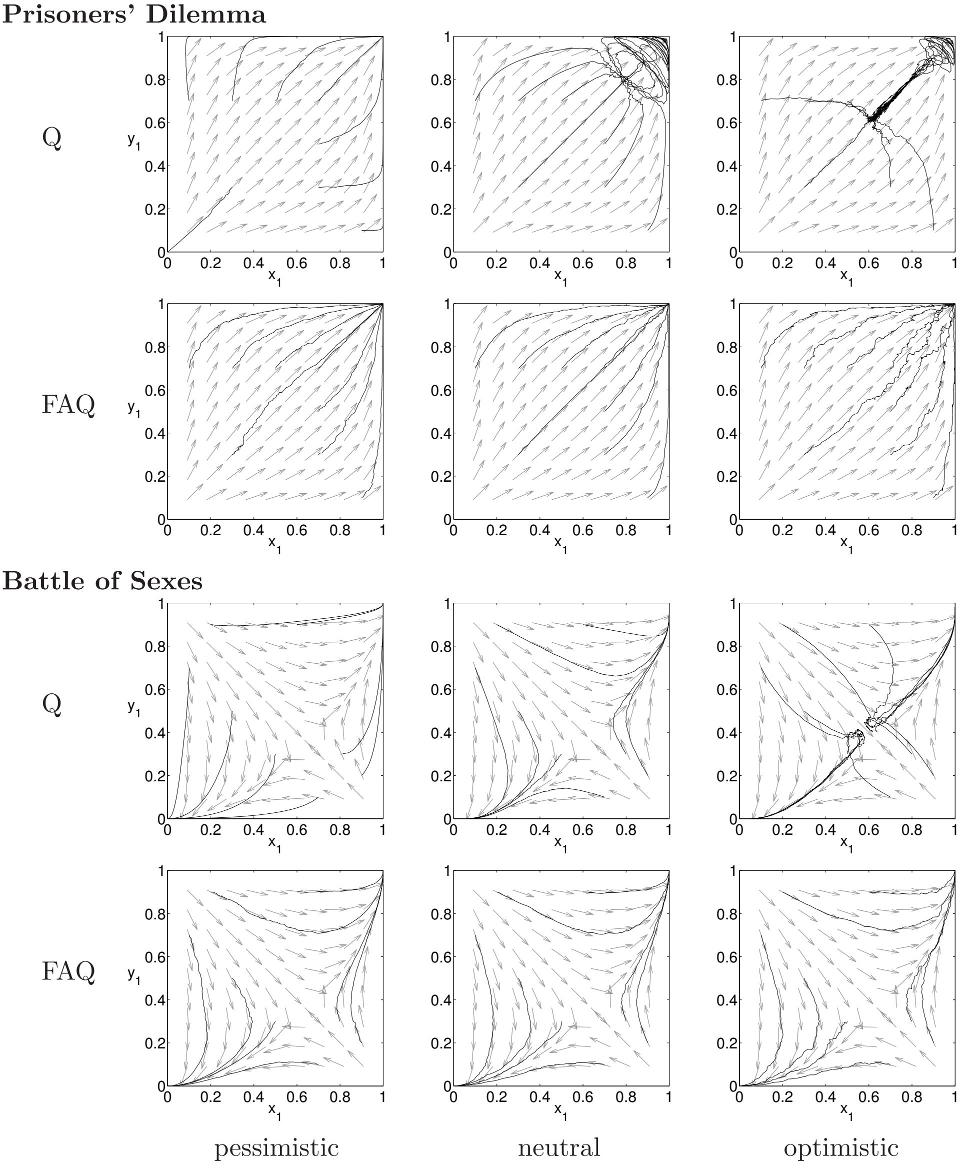


Figure 3.4: Comparison of Q-learning to FAQ-learning with various Q-value initializations in the Prisoners' Dilemma and Battle of Sexes. The Q-values are initialized centered at the minimum (left), mean (center) and maximum (right) possible Q-value given the reward space of the game. The idealized model (arrows) matches the observed FAQ-learning dynamics.

3.2.3 Discussion

The results have shown empirical confirmation of the match between trajectories of the newly proposed FAQ-learning algorithm and its evolutionary prediction, i.e., the difference between trajectories of the stochastic algorithm and the dynamical system are very small. This result is insensitive to the specific values of γ and α, as long as α is reasonably small. Given the Q-value space and a specific temperature τ, the most extreme policy can be computed using the policy generating function given in Equation 2.5. Hence, a temperature τ can be selected such that $x_i \geqslant \beta$ is guaranteed in FAQ-learning, and the algorithm behaves according to the formal FAQ-learning dynamics for the complete range of valid policies under the given temperature. Using analogous derivations as in Section 3.2.1, Frequency Adjusted SARSA (FAS) can be shown to behave equivalently in single-state environments.

Further experiments are required to verify the performance gain in multi-state domains and real applications; the relation between the learning speed $\frac{\alpha\beta}{x_i}$ in FAQ- and α in Q-learning is critical for the speed and quality of convergence that is achieved and needs further investigation. However, the theoretical merits of FAQ-learning are established: FAQ-learning implements the idealized model of Q-learning and yields rational policy improvements in self-play. This result provides a solid basis for the formal analysis of its convergence behavior as presented in the next section.

3.3 Convergence in two-action two-player games

This section further analyzes the behavior of Frequency Adjusted Q-learning (FAQ-learning) in two-agent two-action matrix games. It provides empirical and theoretical support for the convergence of FAQ-learning to attractors near Nash equilibria. The dynamics are evaluated in the three known representative two-agent two-action games: Matching pennies, Prisoners' Dilemma and Battle of Sexes. Results show that Matching-Pennies and Prisoners'-Dilemma type games yield one attractor of the learning dynamics. In contrast, Battle-of-Sexes type games feature one attractor for high exploration (temperature τ), and a supercritical pitchfork bifurcation at a critical temperature, below which there are two attracting and one repelling fixed point. Fixed points in all games approach Nash equilibria as the temperature tends to zero.

The remainder of this section is structured as follows: First, the learning dynamics of FAQ-learning in two-agent two-action matrix games are examined theoretically. Subsequently, simulation experiments that illustrate the learning behavior and convergence near Nash equilibria in three representative games are given. Finally, the main contributions are discussed in relation to previous and ongoing research.

3.3.1 Preliminaries

Recall that FAQ-learning uses the softmax activation function for policy-generation, and a modified Q-learning update rule. The magnitude of each learning step for action i is adjusted by the inverse of the action probability x_i (computed at time t according to Eq. 2.5). FAQ-learning approximates simultaneous action-value estimate updates by increasing the learning steps of less frequently selected actions.

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i}\alpha\left(r_i(t) + \gamma \max_j Q_j(t) - Q_i(t)\right).$$

Tuyls et al. [Tuyls et al., 2003] extended the work on Cross learning and the replicator dynamics [Börgers and Sarin, 1997] to Q-learning. More precisely, they derived the dynamics of the Q-learning process under the simplifying assumption of simultaneous action updates. This analysis yields the following system of differential equations, which according to Section 3.2 precisely describes the FAQ-learning dynamics for a two-player stateless matrix game:

$$\begin{aligned} \dot{x}_i &= x_i\alpha\left(\tau^{-1}\left[e_iAy - xAy\right] - \log x_i + \sum_k x_k \log x_k\right) \\ \dot{y}_j &= y_j\alpha\left(\tau^{-1}\left[xBe_j - xBy\right] - \log y_j + \sum_l y_l \log y_l\right). \end{aligned} \tag{3.3}$$

with x, y the policies, α the learning rate, τ temperature parameter, A, B the payoff matrices, and e_i the i^{th} unit vector. The striking part of this result is that the equations contain a selection part equal to replicator dynamics, and a mutation part. For an elaborate discussion in terms of selection and mutation operators, please refer to the published literature [Tuyls et al., 2006, 2003].

With this model, it is possible to get insight into the learning process, its traces, basins of attraction, and stability of equilibria, by just examining the coupled system of replicator equations and plotting its force and directional fields. An example plot of the dynamics of the game Battle of Sexes is given in Figure 3.5; the corresponding payoff matrix can be found in Figure 3.2.

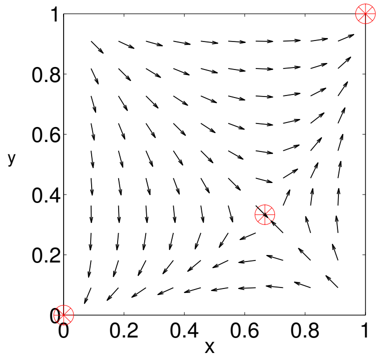


Figure 3.5: The dynamics of FAQ-learning (arrows) with very low exploration in the Battle of Sexes, illustrating that fixed points of the learning process (indicated by ⊗) correspond to Nash equilibria.

3.3.2 Proof of convergence

This section delineates the theoretical support for convergence of FAQ-learning. The dynamical system defined by $\dot{x}$ and $\dot{y}$ in Equation 3.3 yields a number of fixed points, which may be attracting or repelling. Since learning trajectories converge to attractors, the local stability (attracting or repelling) is the main condition that is analyzed.

For notational convenience, I define auxiliary variables a, b and functions K_1, K_2 to simplify the FAQ dynamics $\dot{x}, \dot{y}$ from Equation 3.3. Since only two-action games are considered here, the action index is also dropped for the remainder of this section. In particular, let the row player play policy $(x, 1-x)$ against the column player with policy $(y, 1-y)$:

$$\begin{aligned}
a_1 &= A_{11} - A_{21} \\
a_2 &= A_{12} - A_{22} \\
b_1 &= B_{11} - B_{12} \\
b_2 &= B_{21} - B_{22} \\
h &= (1, -1) \\
hAh^T &= a_1 - a_2 \\
hBh^T &= b_1 - b_2 \\
K_1(x, y) &= \tau_1^{-1}\left[yhAh^T + a_2\right] - \log \frac{x}{1-x} \\
K_2(x, y) &= \tau_2^{-1}\left[xhBh^T + b_2\right] - \log \frac{y}{1-y} \\
\dot{x} &= \alpha x(1-x)K_1(x, y) \\
\dot{y} &= \alpha y(1-y)K_2(x, y).
\end{aligned}$$

At a fixed point, $\dot{x} = \dot{y} = 0$. FAQ-learning with positive exploration parameter τ only covers the open set of policies (x, y) with $x, y \notin \{0, 1\}$, hence $\alpha x(1-x) \neq 0$. As a consequence, $\dot{x} = \dot{y} = 0$ implies $K_1(x, y) = K_2(x, y) = 0$.

The local stability can be analyzed by checking the eigenvalues of the Jacobian matrix $J = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{bmatrix}$ at a fixed point [Hirsch et al., 2004]:

$$\begin{aligned}
\frac{\partial \dot{x}}{\partial x} &= \alpha\left[(1-2x)K_1(x, y) - 1\right] \\
\frac{\partial \dot{x}}{\partial y} &= \alpha x(1-x)\tau_1^{-1}hAh^T \\
\frac{\partial \dot{y}}{\partial x} &= \alpha y(1-y)\tau_2^{-1}hBh^T \\
\frac{\partial \dot{y}}{\partial y} &= \alpha\left[(1-2y)K_2(x, y) - 1\right].
\end{aligned}$$

Since it is established that $K_1(x,y) = K_2(x,y) = 0$ at mixed fixed points, the Jacobian simplifies:

$$J(x,y) = \left[\begin{array}{cc} -\alpha & \alpha x(1-x)\tau^{-1} hAh^T \\ \alpha y(1-y)\tau^{-1} hBh^T & -\alpha \end{array} \right].$$

The eigenvalues can be computed using the quadratic formula:

$$\begin{aligned} \lambda_{1/2} &= -\alpha \pm \frac{1}{2}\sqrt{4\frac{\partial \dot{x}}{\partial y}\frac{\partial \dot{y}}{\partial x} + (-\alpha - (-\alpha))^2} \\ &= -\alpha \pm \sqrt{\frac{\partial \dot{x}}{\partial y}\frac{\partial \dot{y}}{\partial x}} \\ &= -\alpha \pm \alpha\sqrt{x(1-x)y(1-y)\tau_1^{-1} hAh^T \tau_2^{-1} hBh^T}. \end{aligned}$$

Dynamical systems theory has established that fixed points are locally attracting if $\forall\lambda : \mathsf{real}(\lambda) \leqslant 0$ and $\exists\lambda : \mathsf{real}(\lambda) < 0$ [Hirsch et al., 2004]. This fact leads to the following condition for stability, which will be denoted $C(x,y) \leqslant 1$:

$$\begin{aligned} \alpha\left[-1 \pm \sqrt{x(1-x)y(1-y)\tau_1^{-1}\tau_2^{-1} hAh^T hBh^T}\right] &\leqslant 0 \\ -1 \leqslant \sqrt{x(1-x)y(1-y)\tau_1^{-1}\tau_2^{-1} hAh^T hBh^T} &\leqslant 1 \\ C(x,y) = x(1-x)y(1-y)\tau_1^{-1}\tau_2^{-1} hAh^T hBh^T &\leqslant 1. \end{aligned}$$

Since $x, (1-x), y, (1-y), \tau_1, \tau_2$ all are positive, this condition holds independent of x, y if $hAh^T hBh^T \leqslant 0$, leading to eigenvalues with $\mathsf{real}(\lambda) = -\alpha < 0$. In other words, games that satisfy $hAh^T hBh^T \leqslant 0$ have only attracting fixed points. These games already cover all Matching-Pennies type games and some Prisoners'-Dilemma type games.

The following system of equations defines the stability boundary using two conditions for the fixed point, and one for local stability:

$$\begin{aligned} \tau_1 \log\frac{x}{1-x} - a_2 &= yhAh^T \\ \tau_2 \log\frac{y}{1-y} - b_2 &= xhBh^T \\ x(1-x)y(1-y)hAh^T hBh^T &\leqslant \tau_1\tau_2. \end{aligned}$$

This set of equations can be solved numerically for any specific game to obtain fixed points and their stability property. The following general discussion will provide support for convergence in all three classes, especially discussing the characteristic number $hAh^T hBh^T$ associated with each type of game. The two-player two-action normal form games are partitioned into these three classes by conditions on the game-specific auxiliary variables a_1, a_2 and b_1, b_2.

Class 1 Matching Pennies games: I. $a_1a_2 < 0$, II. $b_1b_2 < 0$, and III. $a_1b_1 < 0$.

To link these conditions to the stability property, consider that $hAh^T hBh^T = a_1b_1 - a_1b_2 - a_2b_1 + a_2b_2$. Assumptions I and II imply $a_1a_2b_1b_2 > 0$, hence a_1b_2 and a_2b_1 are either both positive or both negative. Dividing out III, one finds $a_2b_2 < 0$. Assume a_1b_2 is negative, then $a_1b_2a_1a_2 > 0$ leads to the contradiction $a_1^2 < 0$. Since all numbers in the matrix need to be real, it can be concluded that $a_1b_2 > 0$ and $a_2b_1 > 0$. In sum, $hAh^ThBh^T < 0$, which leads to the eigenvalues λ of the Jacobian matrix to have $\mathsf{real}(\lambda) = -\alpha$ as explained above. The fixed point is necessarily attracting in matching pennies games, since $\forall\lambda, \mathsf{real}(\lambda) < 0$.

Class 2 Prisoners' dilemma games: I. $a_1a_2 > 0$ and II. $b_1b_2 > 0$.

Games of this class can have both positive and negative characteristic numbers. Games with $hAh^ThBh^T < 0$ yield necessarily attracting fixed points for the same reason as in Class 1. However, a large number of games of this type have positive characteristic numbers, e.g., for symmetric games $hAh^ThA^Th^T \geqslant 0$. It remains to show that games with III. $(a_1 - a_2)(b_1 - b_2) \geqslant 0$ have attracting fixed points.

From I and II one knows that $[ya_1 + (1-y)a_2] \neq 0$ and $[xb_1 + (1-x)b_2] \neq 0$. This fact implies that there is only one solution to $K_1(x,y) = K_2(x,y) = 0$:

$$\begin{aligned} y\frac{a_1}{\tau_1} + (1-y)\frac{a_2}{\tau_1} &= \log\frac{x}{1-x} \\ x\frac{b_1}{\tau_2} + (1-x)\frac{b_2}{\tau_2} &= \log\frac{y}{1-y}. \end{aligned}$$

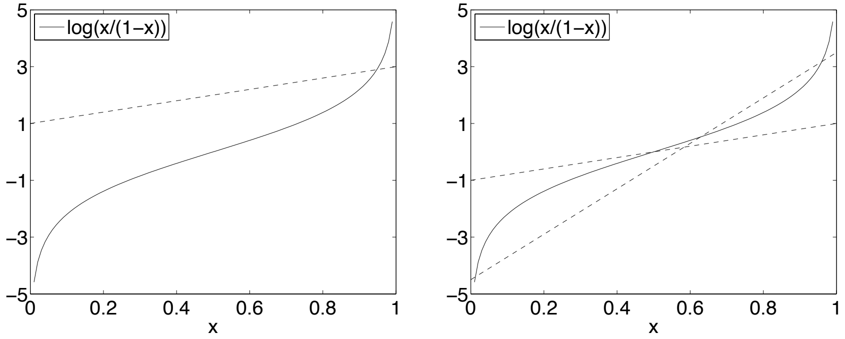
Figure 3.6 plots an example of the first equation. The temperature τ determines the point of intersection between the two lines: If a_1 and a_2 are positive, then $x \to 1$ as $\tau \to 0$. If a_1 and a_2 are negative, then $x \to 0$ as $\tau \to 0$. Equivalent conditions hold for y in relation to b_1 and b_2.

It is trivial to check that the stability condition holds for sufficiently large temperatures. Since $x(1-x)$ goes to zero faster than τ_1 does, and similarly $y(1-y)$ goes to zero faster than τ_2 does, the stability condition $x(1-x)y(1-y)hAh^ThBh^T \leqslant \tau_1\tau_2$ holds for all temperatures $\tau > 0$.

Class 3 Battle of Sexes games: I. $a_1a_2 < 0$, II. $b_1b_2 < 0$, and III. $a_1b_1 > 0$.

The first two conditions imply $a_1a_2b_1b_2 > 0$, hence a_1b_2 and a_2b_1 are either both positive or both negative. Dividing out the third assumption, one finds $a_2b_2 > 0$. Assume a_1b_2 is positive, then $a_1b_2a_1a_2 < 0$ leads to the contradiction $a_1^2 < 0$. Since all numbers in the matrix need to be real, it can be concluded that $a_1b_2 < 0$ and $a_2b_1 < 0$. As a result, the characteristic number $(a_1-a_2)(b_1-b_2) = a_1b_1-a_1b_2-a_2b_1+a_2b_2 > 0$.

From I and II, one knows that $[ya_1 + (1-y)a_2]$ and $[xb_1 + (1-x)b_2]$ both cross zero. Figure 3.6 illustrates the difference between the Prisoners' Dilemma and the Battle of Sexes. It shows the function $\mathsf{log}\frac{x}{1-x}$ and the linear interpolation between $\frac{a_1}{\tau_1}$ and $\frac{a_2}{\tau_1}$. Large values of τ lead to one intersection, while sufficiently small values of τ lead to three intersections and corresponding fixed points. The stability condition $x(1-x)y(1-y)hAh^ThBh^T \leqslant \tau_1\tau_2$ is satisfied for large τ. At the critical temperature $\tau_{\mathtt{crit}}$, the stability condition holds with equality, leading to a supercritical pitchfork bifurcation of the fixed points in τ. Below the critical temperature, two fixed points approach pure Nash equilibria and are stable for the same reasons as the fixed point in the Prisoners' Dilemma. In addition, one fixed point remains mixed, and $x(1-x)$



Prisoners' Dilemma Battle of Sexes

Figure 3.6: Fixed points as intersections of two functions: The Prisoners' Dilemma features one fixed point, because there is exactly one intersection between the linear combination of $\frac{a_1}{\tau_1}$ and $\frac{a_2}{\tau_1}$ with the $\log$ function. The Battle of Sexes yields one or three fixed points, depending on the slope of the linear combination.

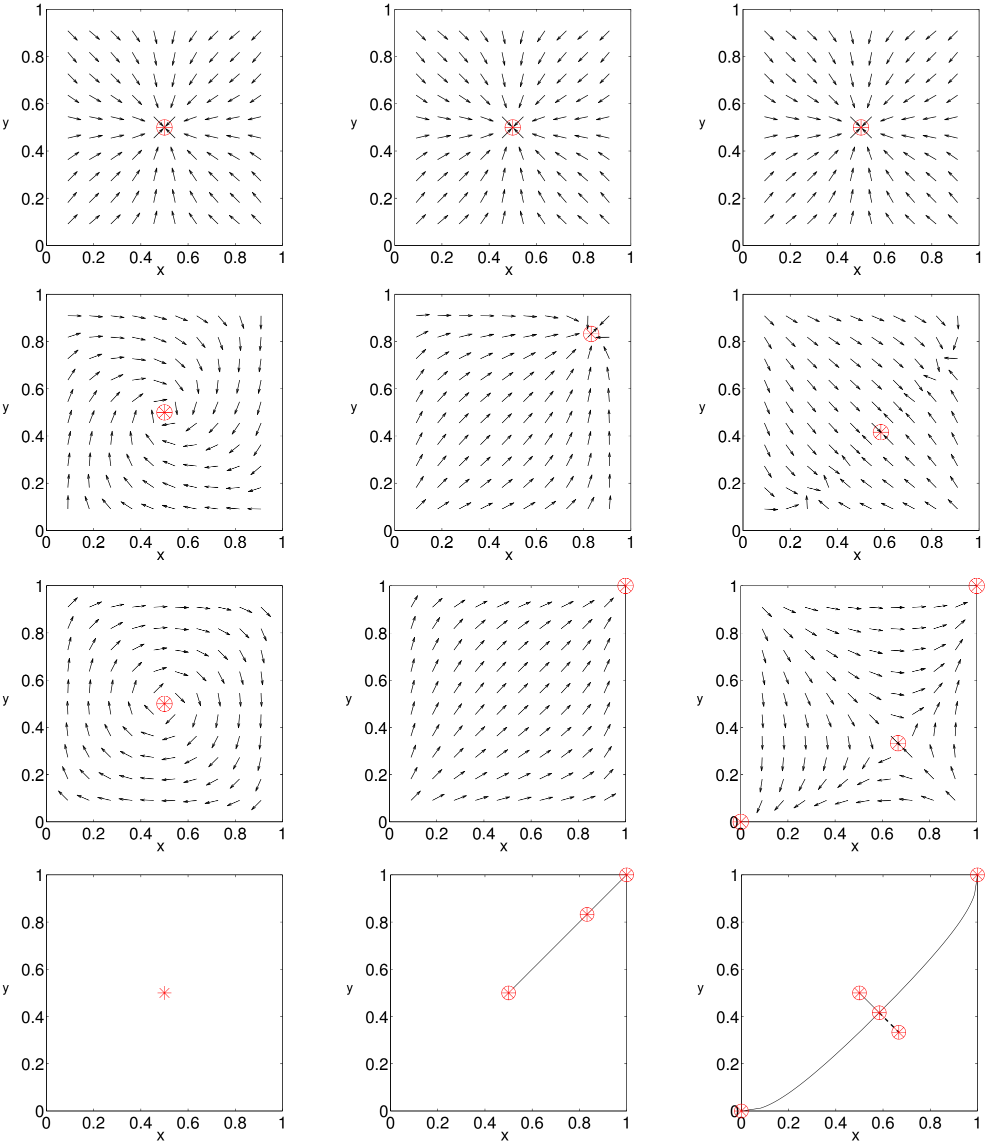
as well as $y(1-y)$ is clearly bound away from zero. As a result, this fixed point is not stable below the critical temperature.

The three discussed classes of two-agent two-action games cover all games in that regime. Hence, it can be concluded that FAQ-learning yields attracting fixed points in all two-agent two-action normal form games.

3.3.3 Experiments

This section illustrates the convergence behavior, and the effect of the exploration parameter τ on the distance of fixed points to Nash equilibria. Each class of two-agent two-action games is represented by one specific game. The payoff bi-matrices (A, B) for Matching Pennies (Class 1), Prisoners' Dilemma (Class 2), and Battle of Sexes (Class 3) are given in Figure 3.2, and repeated in Figure 3.7 for reference. Let the row player play policy $(x, 1-x)$ against the column player with policy $(y, 1-y)$. The Nash equilibria (x, y) of these games lie at $(\frac{1}{2}, \frac{1}{2})$ for the Matching Pennies, $(1, 1)$ for the Prisoners' Dilemma, and at $(0, 0)$, $(1, 1)$, and $(\frac{2}{3}, \frac{1}{3})$ for the Battle of Sexes. The replicator dynamics $(\dot{x}, \dot{y})$ make it possible to determine the coarse location of attractors by inspection. In addition, the fixed points have been computed, and are marked with $\otimes$.

The top three rows show replicator dynamics and the computed fixed points for different temperature parameters τ (first row $\tau = \infty$, second row $\tau = 0.72877$, third row $\tau = 0$). The fixed points move between these discrete values for τ as indicated by the lines of the fourth row. For reference, all fixed points computed for the discrete values are also marked in the fourth row.



$$\begin{array}{c|cc} & \text{H} & \text{T} \\ \text{H} & 2,0 & 0,2 \\ \text{T} & 0,2 & 2,0 \end{array}$$

Matching Pennies

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

Prisoners' Dilemma

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

Battle of Sexes

Figure 3.7: Replicator dynamics (arrows) and fixed points ($\otimes$) for $\tau \in \{\infty, 0.72877, 0\}$ (first three rows). Fourth row shows trajectories of fixed points as temperature is decreased. All fixed points are attracting, except for the mixed fixed point that tends to $(\frac{2}{3}, \frac{1}{3})$ after bifurcation (indicated with a dashed line).

The dynamics of FAQ-learning are independent of the game when the exploration parameter τ tends to infinity. This property is expected, since in that case no exploitation of the payoff information is used. For finite temperatures, the three games exhibit very different behavior. However, the fixed points approach the Nash equilibria of the games in the limit of $\tau \to 0$ in all three cases (see row 3).

The Matching-Pennies game yields one mixed equilibrium, which is also an attracting fixed point of the FAQ-learning dynamics for any positive τ. In the limit $\tau \to 0$, the fixed point's stability weakens to Lyapunov stability (points that start close will stay close to the fixed point, but not necessarily converge to it). This behavior may be conjectured from the inspection of the dynamics, and can be confirmed using the stability analysis from Section 3.3.2.

The Prisoners'-Dilemma game yields one pure equilibrium, and one mixed fixed point that is always attracting. The lower the temperature τ, the closer the fixed point moves towards the equilibrium. It is also stable in the limit $\tau \to 0$.

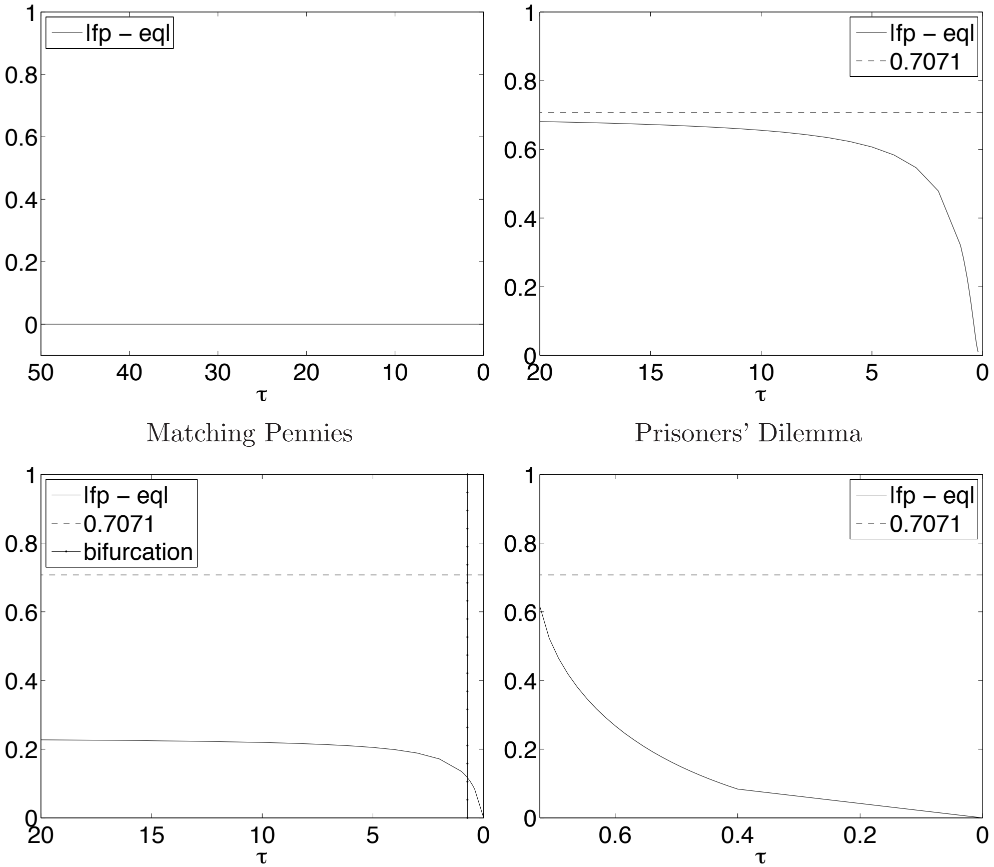
The Battle-of-Sexes game yields three Nash equilibria. However, for high values of τ, it only yields one attracting fixed point that moves from $(\frac{1}{2}, \frac{1}{2})$ toward the mixed equilibrium $(\frac{2}{3}, \frac{1}{3})$. This fixed point splits in a supercritical pitchfork bifurcation at the critical temperature $\tau_{crit} \approx 0.72877$ and at position $(x, y) \approx (0.5841, 0.4158)$. For low temperatures $\tau < \tau_{crit}$, this game yields three fixed points that move closer to the corresponding equilibria as τ is decreased. The two fixed points moving toward the pure equilibria $(0, 0)$ and $(1, 1)$ are attracting, and the third one moving toward $(\frac{2}{3}, \frac{1}{3})$ is repelling.

The relation between the exploration parameter τ of FAQ-learning and the distance between fixed points and Nash equilibria is closely examined in Figure 3.8. It shows that the distance is constant zero for Matching Pennies, and monotonically decreasing toward zero for the other two games. Notably, the two emerging fixed points in the Battle of Sexes result in the same distance plot, due to a certain symmetry of the game.

In sum, FAQ-learning converges to fixed points in the three representative games Matching Pennies, Prisoners' Dilemma and Battle of Sexes. In addition, these fixed points can be moved arbitrarily close to the Nash equilibria of these games by choosing an exploration parameter τ close to zero.

3.3.4 Discussion

This section has proven that FAQ-learning converges to fixed points which approach Nash equilibria as exploration is decreased. Since FAQ-learning only differs from Q-learning in the learning speed modulation of individual actions, this result can probably be extended to classical Q-learning. However, this extension is not straight-forward, since the dynamics of Q-learning are higher-dimensional than those of FAQ-learning. This fact prevents the evaluation of the Jacobian in the two-dimensional policy space of two-agent two-action games, and requires the application of convergence analysis to the four-dimensional Q-value space.



Matching Pennies

Prisoners' Dilemma

Battle of Sexes, distance to mixed equilibrium $(\frac{2}{3}, \frac{1}{3})$

Battle of Sexes, distance to pure equilibria $(0, 0)$, $(1, 1)$

Figure 3.8: The distance between fixed points (fp) of FAQ-learning dynamics and Nash equilibria (eq) as a function of the exploration parameter τ. As $\tau \to 0$, the distance $|\mathrm{fp} - \mathrm{eq}| \to 0$ as well.

Other authors have used the approach to describe multi-agent reinforcement learning by a dynamical system with infinitesimal learning rates [Babes et al., 2009; Gomes and Kowalczyk, 2009; Tuyls et al., 2006, 2003; Wunder et al., 2010]. However, these related results do not provide convergence guarantees. After publication of this proof of convergence in a workshop [Kaisers and Tuyls, 2011], an alternative proof of convergence starting from equivalent premises has been published by other authors [Kianercy and Galstyan, 2012]. The methodology deployed for the analysis within this section is gaining momentum in the multi-agent learning literature.

3.4 Summary

The contributions of this chapter can be summarized as follows: The deviation of Q-learning from its evolutionary model has been analyzed and explained. The policy dynamics of classical Q-learning cannot be described in a self-consistent way. Based on the new insights, FAQ-learning has been introduced and is shown to comply with the idealized model for an arbitrarily large part of the policy space, thereby exhibiting game theoretically more desirable behavior than Q-learning. Next, it is shown theoretically that fixed points of FAQ-learning are attracting in Matching-Pennies and Prisoners'-Dilemma type games, and that a supercritical pitchfork bifurcation occurs in Battle-of-Sexes type games. In addition, representative example games of each category demonstrate that fixed points approach Nash equilibria if exploration is decreased, and illustrate the bifurcation of fixed points in the Battle of Sexes.

These results contribute to the study of multi-agent learning by deepening the understanding of convergence properties of independent reinforcement learning in multi-agent settings. The method has been shown to work in the policy space, and naturally extends to the Q-value space, which makes it possible to generalize insights to classical Q-learning. By doing so, future work can strengthen the theoretical guarantees and their impact to a wide array of applications.

4

Extending the dynamical systems framework

The formal analysis of multi-agent reinforcement learning is challenging and requires a more complex approach than single-agent learning. The merits of dynamical systems as a framework for multi-agent learning have been demonstrated in the previous chapter, e.g., by providing a proof of convergence. Here, this framework is extended in three ways to allow application to a wider range of problems. First, exploration in single-agent learning is commonly decreased over time, but previous evolutionary models only capture variations of Q-learning with a constant exploration rate. As a resolution, the next section derives a model of Frequency Adjusted Q-learning (FAQ-learning) with a time-dependent exploration rate. This section is based on published material [Kaisers et al., 2009]. Second, the dynamical systems framework is well-established in single-state games, and Q-value dynamics and policy dynamics for FAQ-learning and SARSA are extended to stochastic games in Section 4.2. In addition, an alternative approach to stochastic games is sketched as suggested in prior work [Hennes et al., 2010]. Third, cooperative learning may get stuck in local optima. The concept of leniency has been used in related work to increase probability of convergence to the global optimum and is well-captured in the evolutionary model, as shown in Section 4.3 [Bloembergen et al., 2011]. A summary of the findings concludes the chapter.

4.1 Time-dependent exploration rates in Q-learning

Q-learning is often used with decreasing exploration [Watkins and Dayan, 1992]. However, the idealized model presented in Section 2.5.3 assumes a constant exploration rate. This section introduces an extension of the idealized model of Q-learning, providing the learning dynamics of FAQ-learning with exploration that may vary over time, and compares the newly derived dynamics to Cross learning using experiments in simulated auctions.

4.1.1 The time derivative of softmax activation

The idealized model of Q-learning is based on the time derivative of the softmax activation function assuming a constant temperature parameter τ. Here, the derivative based on a temperature function $\tau(t)$ is given for single-state games. As before, $Q_i(t)$ denotes the Q-value of action i at time t with the notation for state-dependence dropped, and recall $x_i(Q_i(t), \tau(t))$ denotes the probability of selecting action i as a function of Q-values and the exploration parameter. In the following equations, the notation for dependence on t is dropped for the sake of readability. First, the softmax activation function can be decomposed into f_i and g:

$$x_i(Q, \tau) = \frac{e^{\tau^{-1} Q_i}}{\sum_j e^{\tau^{-1} Q_j}} \hat{=} \frac{f_i}{g}.$$

The policy change $\dot{x}_i(t)$ can be computed by applying the quotient rule, which makes use of the derivatives of f_i and g:

$$\begin{aligned}
\frac{df_i}{dt} &= \frac{d}{dt}\left(\tau^{-1} Q_i\right) e^{\tau^{-1} Q_i} \\
&= \left(-\tau^{-2}\frac{d\tau}{dt} Q_i + \tau^{-1}\frac{dQ_i}{dt}\right) f_i \\
\frac{dg}{dt} &= \sum_j \left(-\tau^{-2}\frac{d\tau}{dt} Q_j + \tau^{-1}\frac{dQ_j}{dt}\right) e^{\tau^{-1} Q_j} \\
&= g \sum_j x_j \left(-\tau^{-2}\frac{d\tau}{dt} Q_j + \tau^{-1}\frac{dQ_j}{dt}\right).
\end{aligned}$$

Next, the application of the quotient rule reads as follows:

$$\begin{aligned}
\frac{dx_i}{dt} &= \frac{\frac{df_i}{dt} g - f_i \frac{dg}{dt}}{g^2} \\
&= \frac{\left(-\tau^{-2}\frac{d\tau}{dt} Q_i + \tau^{-1}\frac{dQ_i}{dt}\right) f_i g - f_i g \sum_j x_j \left(-\tau^{-2}\frac{d\tau}{dt} Q_j + \tau^{-1}\frac{dQ_j}{dt}\right)}{g^2} \\
&= x_i \left[\left(-\tau^{-2}\frac{d\tau}{dt} Q_i + \tau^{-1}\frac{dQ_i}{dt}\right) - \sum_j x_j \left(-\tau^{-2}\frac{d\tau}{dt} Q_j + \tau^{-1}\frac{dQ_j}{dt}\right)\right] \\
&= x_i \tau^{-1} \left[\frac{dQ_i}{dt} - \tau^{-1}\frac{d\tau}{dt} Q_i - \sum_j x_j \left(\frac{dQ_j}{dt} - \tau^{-1}\frac{d\tau}{dt} Q_j\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&= x_i\tau^{-1}\left[\frac{dQ_i}{dt} - \tau^{-1}\frac{d\tau}{dt}Q_i - \sum_j x_j\frac{dQ_j}{dt} + \sum_j x_j\tau^{-1}\frac{d\tau}{dt}Q_j\right] \\
&= x_i\tau^{-1}\left[\frac{dQ_i}{dt} - \sum_j x_j\frac{dQ_j}{dt} + \frac{d\tau}{dt}\sum_j x_j\tau^{-1}\left(Q_j - Q_i\right)\right].
\end{aligned}
$$

The dependence on Q-values can be eliminated using the following equality:

$$
\begin{aligned}
\tau^{-1}\left(Q_j - Q_i\right) &= \log e^{\tau^{-1}Q_j} - \log e^{\tau^{-1}Q_i} \\
&= \log\frac{f_j}{g}\frac{g}{f_i} \\
&= \log\frac{x_j}{x_i} \\
&= \log x_j - \log x_i.
\end{aligned}
$$

Substituting this equality, and simplifying using $\sum_j x_j = 1$, the final form in vector notation reads:

$$
\frac{dx_i}{dt} = x_i\tau^{-1}\left[\frac{dQ_i}{dt} - \sum_j x_j\frac{dQ_j}{dt} - \frac{d\tau}{dt}\left(\log x_i - x\log x^T\right)\right]. \tag{4.1}
$$

The derivative of softmax activation depends on the Q-value derivative of all actions. That Q-derivative varies depending on the Q-learning update rule that is used. Equation 4.1 can be used to plug in Q-value derivatives of Q-learning, SARSA, or FAQ-learning. Section 4.2 elaborates on multi-state extensions. The following section will specify the FAQ-learning dynamics, which can be completely reduced to the policy space.

4.1.2 Frequency Adjusted Q-learning with varying exploration

FAQ-learning has been analyzed in detail in Chapter 3. Recall Equation 3.2, which gives the Q-value change $\frac{dQ_i}{dt}$ for FAQ-learning:

$$
\frac{dQ_i}{dt} = \alpha\left(E\left[r_i(t)\right] + \gamma\max_j Q_j(t) - Q_i(t)\right).
$$

This Q-value change has previously been used to compute the dynamical system assuming a constant temperature. Here, it is substituted in the softmax activation derivative with a time-dependent temperature function (Equation 4.1), which leads to the dynamics of FAQ-learning with varying exploration. The notation for time dependence of $x_i(t), r_i(t)$ and $\tau(t)$ are dropped for readability:

$$
\frac{dx_i}{dt} = \quad x_i\tau^{-1}\left[\alpha\left(E\left[r_i\right] + \gamma\max_k Q_k - Q_i\right) - \sum_j x_j\alpha\left(E\left[r_j\right] + \gamma\max_k Q_k - Q_j\right)\right]
$$

$$
\begin{aligned}
& -x_i\tau^{-1}\frac{d\tau}{dt}\left(\log x_i - x\log x^T\right) \\
= \quad & x_i\alpha\tau^{-1}\left[E\left[r_i\right] - Q_i - \sum_j x_j E\left[r_j\right] + \sum_j x_j Q_j\right] \\
& -x_i\tau^{-1}\frac{d\tau}{dt}\left(\log x_i - x\log x^T\right) \\
= \quad & x_i\alpha\tau^{-1}\left[E\left[r_i\right] - \sum_j x_j E\left[r_j\right] + \sum_j x_j\left(Q_j - Q_i\right)\right] \\
& -x_i\tau^{-1}\frac{d\tau}{dt}\left(\log x_i - x\log x^T\right) \\
= \quad & x_i\alpha\tau^{-1}\left[E\left[r_i\right] - \sum_j x_j E\left[r_j\right] + \sum_j x_j\tau\left(\log x_j - \log x_i\right)\right] \\
& -x_i\tau^{-1}\frac{d\tau}{dt}\left(\log x_i - x\log x^T\right),
\end{aligned}
$$

which leads to the simplified form:

$$
\frac{dx_i}{dt} = x_i\frac{\alpha}{\tau}\left[E\left[r_i\right] - \sum_j x_j E\left[r_j\right]\right] - x_i\left(\alpha + \frac{1}{\tau}\frac{d\tau}{dt}\right)\left(\log x_i - x\log x^T\right).
$$

In this form, several key features become apparent. First, the dynamics feature an exploitation part equivalent to the replicator dynamics, and an exploration part related to information gain. The exploitation term vanishes as exploration increases to infinity. For the formal analysis of infinitesimal learning rates, consider the equivalent form using $\beta = \frac{\alpha}{\tau}$:

$$
\frac{dx_i}{dt} = \underbrace{x_i\beta\left[E\left[r_i\right] - \sum_j x_j E\left[r_j\right]\right]}_{\text{replicator dynamics}} \underbrace{- x_i\left(\tau\beta + \frac{1}{\tau}\frac{d\tau}{dt}\right)\overbrace{\left(\log x_i - x\log x^T\right)}^{\text{information gain}}}_{\text{exploration}}. \tag{4.2}
$$

The exploration term combines effects of exploration, vanishing as the exploration parameter τ approaches zero, and effects of change in exploration. Given a constant exploration rate, the equation degenerates to the previously established idealized model (Equation 2.7). In fact, if learning is allowed infinite time, the derivative $\frac{d\tau}{dt}$ can be made arbitrarily small. The learning process will therefore asymptotically be the only driving force of the policy adaptation process and the degenerate model may be applied. However, time is finite for all practical applications and the extended model is required for designing appropriate temperature functions.

4.1.3 Designing appropriate temperature functions

Exploration is used to overcome local optima and increase the probability of converging to global optima. However, an appropriate function of exploration over time needs to be chosen. This section discusses the resulting challenges using a one-population model of three strategies. The next section presents simulation experiments with an equivalent model in the domain of auctions.

For single-agent learning, or more generally for stochastic approximation algorithms, theory prescribes specific constraints for appropriate learning rates [Tsitsiklis, 1994]. More specifically, $\sum_{t=1}^{\infty} \alpha(t) = \infty$ and $\sum_{t=1}^{\infty} \alpha(t)^2 < C$ for some constant C. These constraints give guidance on how to select temperature functions to prove convergence of single-agent learning. In essence, the convergence relies on the fact that learning steps become smaller over time and thus stationary eventually. For multi-agent learning, the demands are more complex, since optimality depends on the opponents behavior and is better defined as a best response. General proofs of convergence are hard to obtain, because an arbitrarily small change in the opponent's policy may induce an arbitrarily large change in the best response. More specifically, consider an opponent playing strategy $y \in [0, 1]$, and let the payoff to the learning agent be defined as $f_1(y) = y$ and $f_2(y) = 1 - y$ for action one and two, respectively. Then, an arbitrarily small change δ of the opponent from $y = \frac{1}{2} + \delta$ to $y = \frac{1}{2} - \delta$ changes the best reply from playing purely action 1 to playing it not at all. As a result, even in self-play (i.e., learning algorithms only face opponents of their own type), where all agents use the same decreasing learning steps, the decrease of the learning step size may not happen too fast, as it limits the attainable policy change for that agent. However, to reach the best response, this policy change may need to be large. Therefore, multi-agent learning requires new guidelines on designing appropriate temperature functions, possibly only reaching stability (or at least ϵ-stability) if learning never ceases.

Consider the simplest example of multi-agent learning: self-play with shared memory. This kind of learning can be modeled using one-population dynamics. For simplicity, the examples in this section are based on three strategies; the population distribution $x = (x_1, x_2, x_3)$ can be visualized in a simplex. Each corner represents a pure population of only one of the strategies, boundaries mix between two strategies, and an interior point represents a mix of all strategies proportional to the position of the point. The population changes according the dynamics $\dot{x} = (\dot{x}_1, \dot{x}_2, \dot{x}_3)$, where maintaining the population distribution $\sum_i x_i = 1$ and $\forall i : 0 \leqslant x_i \leqslant 1$ implies $\sum_i \dot{x}_i = 0$.

As described in the previous section, the FAQ-learning dynamics with varying temperature (Equation 4.2) can be decomposed into several terms: the replication dynamics RD, which encode exploitation, and an information gain term IG, which is scaled by exploration $\beta\tau$ and relative change in exploration $\frac{1}{\tau}\frac{d\tau}{dt}$. These force fields of learning and exploration are vector fields that simultaneously influence the policy:

$$\frac{dx_i}{dt} = \mathsf{RD} - x_i \left(\tau\beta + \frac{1}{\tau}\frac{d\tau}{dt} \right) \mathsf{IG}.$$

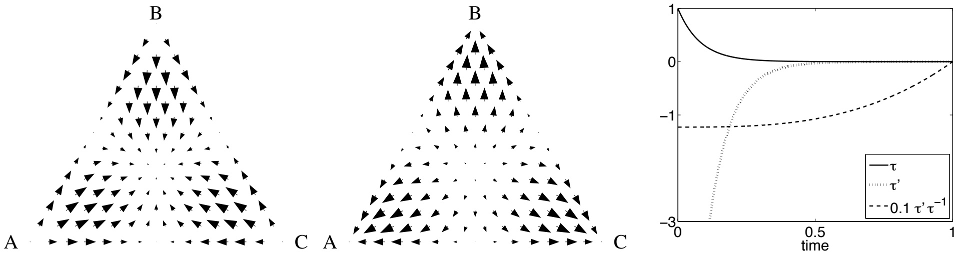
The exploration term can be further decomposed into $x_i\mathsf{IG}$, which is only a function of x, and $-\left(\tau\beta + \frac{1}{\tau}\frac{d\tau}{dt}\right)$, which depends on t and determines the scale and sign of the resulting force. The first term can be computed for a grid of points in the simplex to give an intuition of the resulting force, and since it only depends on x it always has one of the two forms depicted in Figure 4.1 (a), either for increasing or for decreasing the temperature—it only varies in magnitude. These plots feature arrows in the direction of $\dot{x}$ which are scaled proportionally to $|\dot{x}|$. Figure 4.1 (b) plots an example temperature function, derived below, over the interval $[0, 1]$ and shows how the term $\frac{1}{\tau}\frac{d\tau}{dt}$ ensures that the influence of temperature decrease on policy change vanishes over time. Furthermore, it reveals that the derivative of the temperature makes it possible to balance the two terms, i.e., if the temperature decrease is spread over a longer time, learning can compensate more easily because $\dot{\tau}$ is smaller.

Next, a temperature function is designed for the interval $[0, 1]$ that exerts a controlled force on the learning process. To minimize the impact of the temperature decrease on the learning process, the fraction $\frac{1}{\tau}\frac{d\tau}{dt}$ should approach zero when time approaches 1, e.g., $\frac{1}{\tau}\frac{d\tau}{dt} = (b - dct^{d-1})$, where b, c, d are calibration parameters, which can be achieved with an exponential function:

$$
\begin{aligned}
\tau(t) &= ae^{bt-ct^d} \\
\dot{\tau}(t) &= (b - dct^{d-1})ae^{bt-ct^d} \\
\text{such that } \frac{1}{\tau}\frac{d\tau}{dt} &= (b - dct^{d-1}).
\end{aligned}
$$

Initially, learners should explore, and over time they should become more rational. The following parameters calibrate this function such that $\tau(0) = \tau_{\max}$ and $\tau(1) = \tau_{\min}$:

$$
\begin{aligned}
a &= \tau_{\max} \\
b &= \log\left(\frac{\tau_{\min}}{\tau_{\max}}\right) + c \\
c &= -\log\left(\frac{\tau_{\min}}{\tau_{\max}}\right)(1-d)^{-1}
\end{aligned}
$$



increasing τ decreasing τ

(a) Force fields (b) Magnitude

Figure 4.1: The force field $x_i x_i^{IG}$ that exploration and a change thereof exhibits on the policy, and scalar functions over time τ and $\frac{1}{\tau}\frac{d\tau}{dt}$ that determine the force field's magnitude.

$$\tau_t = \tau_{\max} e^{\log\left(\frac{\tau_{\min}}{\tau_{\max}}\right)\left(t - t\log\left(\frac{\tau_{\min}}{\tau_{\max}}\right)(1-d)^{-1} - t^{d}(1-d)^{-1}\right)}$$

$$\tau_t = \tau_{\max} \left(\frac{\tau_{\min}}{\tau_{\max}}\right)^{t+(1-d)^{-1}\left(t\frac{\tau_{\min}}{\tau_{\max}}+t^{d}\right)} .$$

The remaining parameter d determines the curvature of the force decrease and is set to $d = 4$ for the experiments below. Temperature is decreased from $\tau_{\max} = 1$ to $\tau_{\min} = 0.0001$. This function corresponds to the plot in Figure 4.1 (b). However, the experiments stretch this function on the interval $[0, 4]$ to decrease the strength of the resulting force, i.e. $\tilde{\tau}(t) = \tau(\frac{1}{4}t)$.

4.1.4 Experiments in auctions

Here, the classical replicator dynamics, which do not bear a term for exploration, are compared to constant exploration rates and the newly derived model with varying exploration. In particular, traders learn the probabilistic mix of three predefined trading strategies that yields the highest expected profit in a population of traders that choose between these strategies. This study extends previous research in auctions by Phelps et al. [Phelps et al., 2006, 2004], which only considered models of Cross learning. Results show the increased basin of attraction for the global optimum given variable exploration and underline the importance of modeling exploration in multi-agent reinforcement learning.

Auctions provide a variety of markets with high impact on today's economy, e.g., stock exchanges and online peer-to-peer trading platforms. The individual trader seeks to maximize its profit, while the competition conceals their strategies. Hence, auctions naturally fit the model of multi-agent systems in which reinforcement learning can be used to maximize payoff. Recent research has applied models of the replicator dynamics, which are linked to the simple reinforcement learner Cross learning, to foster the understanding of rational behavior in auctions [Phelps et al., 2010a,b]. However, the myopic rationality of Cross learning lacks exploration, which is an essential trait of learning. In this section, an experiment based on simulated auctions shows qualitative differences in the asymptotic behavior of rational reinforcement learning versus explorative learning.

Auction setup

Auctions provide a market for traders to exchange goods for money. Buyers and sellers place offers, bids and asks respectively, to indicate their intention to trade at a certain price. The clearing house auction considered here proceeds in rounds and polls offers from each trader each round. When all offers are collected, an equilibrium price is established based on the available offers such that demand meets supply at this price. It is set to the average of the two offers that define the range of possible equilibrium prices, i.e., the lowest bid and the highest ask that can be matched in the equilibrium. The good to be traded may have a different private value for each trader. The difference between the transaction price and the private value of the trading agent determines the agent's profit, assuming that buyers will not buy above and sellers will not sell below their private value.

A multitude of trading strategies has been devised to derive the next offer, possibly exploiting the knowledge about offers and transactions that were observed in previous rounds. The experiments below are based on the same auction setup as Kaisers et al. [Kaisers et al., 2008]: a simulated continuous double auction with the three trading strategies Modified Roth-Erev (MRE), Zero Intelligence Plus (ZIP) and Gjerstad and Dickhaut (GD). Roth and Erev devised a reinforcement-learning model of human trading behavior [Erev and Roth, 1998], which is modified to perform in a clearing house auction as *Modified Roth-Erev* (MRE) [Nicolaisen et al., 2001]. MRE is evaluated in competition to *Gjerstad and Dickhaut* (GD) and *Zero Intelligence Plus* (ZIP). GD maximizes the expected profit by computing the profit and probability of leading to a transaction for a set of relevant prices [Gjerstad and Dickhaut, 1998]. ZIP places stochastic bids within a certain profit margin, which is lowered when a more competitive offer was rejected and increased when a less competitive offer was accepted [Cliff, 1997; Cliff and Bruten, 1997, 1998]. A more detailed survey of these strategies can be found [Parsons et al., 2005; Phelps et al., 2004].

Evaluation methodology

Given a set of available trading strategies, it is of high interest which strategy is *best* in the sense that it yields the highest expected payoff. However, this question cannot be answered in general as the performance of a trading strategy is highly dependent on the competition it faces [Rust et al., 1993]. Walsh et al. [Walsh et al., 2002] have proposed a *heuristic payoff table* to capture the average profit of each trading strategy for all possible mixtures of strategies in the competition of a finite number of n traders. Each predefined trading strategy is taken as an atomic action of a symmetric normal form game. Assuming all agents update their policy according to the Q-learning model, the average behavior of the population can be described by the derived model.

The experiments are based on a heuristic payoff table for a clearing house auction with $n = 6$ traders who may play a probabilistic mixture of the three strategies ZIP, MRE and GD. The distribution of n agents on k pure strategies is a combination with repetition, hence a heuristic payoff table requires $\binom{n+k-1}{n}$ rows. Each row yields a *discrete profile* $N = (N_1, \ldots, N_k)$ telling exactly how many agents play each strategy. The payoffs of these discrete profiles can be measured in many domains, e.g., in simulated auctions. However, measurements are insufficient to capture the payoff to strategies that are not present, i.e., whenever $N_i = 0$ then $U_i(N)$ is unknown for that discrete profile. Table 4.1 shows an excerpt of the heuristic payoff table computed from simulated auctions, indicating unknown payoffs with a dash.

To approximate the payoff for an arbitrary mix of an infinite population, a weighted average is computed from the payoffs that are listed in the heuristic payoff table. More specifically, n traders are drawn from the infinite population according to its distribution. As introduced in Section 2.4.3, this drawing determines the probability of each finite population distribution to occur and can be computed exactly. Let the

Table 4.1: An excerpt of the heuristic payoff table computed for a clearing house auction with 6 agents and the three strategies ZIP, MRE and GD.

N_{ZIP}	N_{MRE}	N_{GD}	u_{ZIP}	u_{MRE}	u_{GD}
6	0	0	99	-	-
5	1	0	97	100	-
	⋮			⋮	
0	1	5	-	43	79
0	0	6	-	-	79

set of all discrete profiles be denoted as $\nu = \{(n, 0, \ldots, 0), \ldots, \ (0, \ldots, 0, n)\}$ and let $\mu_i = \{N \in \nu | N_i = 0\}$ be the set of profiles where strategy i is not played. Furthermore, let $\bar{\mu_i} = \{N \in \nu | N_i \neq 0\}$ be the complement of μ_i. The expected payoff can be computed from the heuristic payoff table:

$$f_i(x) = \frac{\sum_{N \in \bar{\mu_i}} u_i(N) \cdot \binom{n}{N_1, \ldots, N_k} \cdot x_1^{N_1} \cdot \ldots \cdot x_k^{N_k}}{1 - \sum_{N \in \mu_i} \binom{n}{N_1, \ldots, N_k} \cdot x_1^{N_1} \cdot \ldots \cdot x_k^{N_k}}.$$

The normalization in the denominator compensates for unknown payoffs.

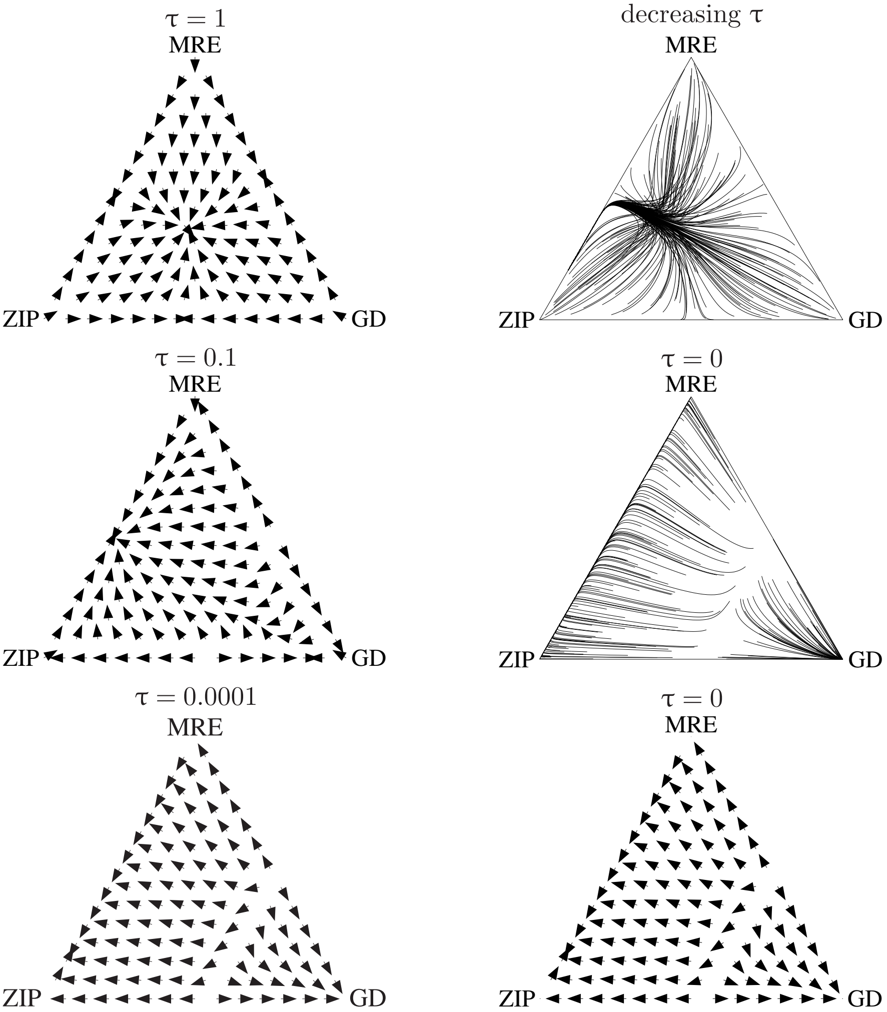
Results

The resulting dynamics can be visualized in a force field plot as in Figure 4.2, where the arrows indicate the direction and strength of change. It can be observed that the selection-mutation model for Q-learning converges to the selection model as τ approaches zero.

The force field plots deliver a snapshot of the direction of population change at a certain time. Since the temperature depends on time, the learning dynamics smoothly change from those depicted for $\tau = 1$ to those depicted for $\tau = 0.0001$. Furthermore, the policy is not only subject to the forces of learning but rather to a linear combination of the forces of learning as in Figure 4.2 and temperature change as in Figure 4.1.

A trajectory plot shows the actual evolution from one or more initial populations over time. It is a discrete approximation of the continuous dynamics, making it possible to analyze the convergence of the initial policy space computationally. Given policy $x_t = (x_{t,ZIP}, x_{t,MRE}, x_{t,GD})$ at time t, these plots are generated from the dynamics defined in Equation 4.2.

Figure 4.2 (b) shows the convergence of 200 example trajectories in the selection model and in the selection-mutation model with decreasing temperature. Each trajectory represents a learning process from a certain initial policy. The policy is not only subject to learning updates but rather to a linear combination of the forces of learning as in Figure 4.2 and temperature change as in Figure 4.1. It can be observed that the convergence behavior of the selection model that inherently features a fixed temperature is completely captured by the snapshot in the directional field plot. The



(a) Directional field plot

(b) Trajectories with uniform starting positions and Cross learning model

Figure 4.2: (a) Replicator dynamics for a set of fixed temperatures and (b) trajectories of the selection model for Cross learning compared to the new model for Q-learning with a varying exploration rate as in Equation 4.2.

selection-mutation model on the other hand features a continuously changing force field and cannot be captured by inspection of the directional and force field plots.

An analysis of 1000 trajectories with uniformly sampled initial policies showed the following convergence: In the selection model, 25.1% converged to the pure profile $(0, 0, 1)$ with payoff 78.51 and 74.9% converged to the mixed profile $(0.813, 0.187, 0)$ with payoff 97.27. These yield an overall expected asymptotic payoff of 92.56 for the selection model given a uniform starting population. In contrast, 100% of the strategy space converges to $(0.811, 0.189, 0)$ with an expected payoff of 97.25 in the mutation model. The results imply that a population of agents that utilize the exploration scheme to overcome local optima may obtain a higher expected payoff than a population of myopic, absolutely rational learners.

4.1.5 Discussion

This model has a strong explanatory power and a clear interpretation. E.g., when the temperature decreases too fast it may make the force of temperature decrease stronger than learning and lead to strange convergence behavior to pure policies. This hypothetical situation matches algorithmic observations, in particular when the initial Q-values are outside the convex combination of values from the reward range.

The contributions of this section are two-fold: On the one hand, the evolutionary model of Q-learning has been extended to account for a varying exploration rate. On the other hand, a case study in the domain of auctions has demonstrated that this model may deliver qualitatively different results, going beyond rational learners and considering a more complex model of learning, which may lead to global rather than local optima. An appropriate temperature function has been designed for this specific case study. However, gaining insight into the dynamics of games, especially if time-dependent, remains a challenging problem. Chapter 5 introduces a new perspective on multi-agent learning dynamics as a new tool to design time-dependent parameters.

4.2 Learning dynamics in stochastic games

So far, the analysis of multi-agent learning dynamics has been applied to the most tractable environments, namely single-state games. Now that Chapter 3 has established a proof of convergence for FAQ-learning and the single-state dynamics have been extended to account for varying exploration rates, it is possible to provide a basis for tackling multi-state environments, here modeled as stochastic games. Therefore, this section first describes the Q-value dynamics for multi-state games, and then extends the policy dynamics of FAQ-learning to multiple states. In addition, the theory is extended to a dynamical systems model of SARSA, which is a variant of Q-learning with a slightly different update scheme. Finally, an alternative approach to stochastic games based on networks of learning algorithms is sketched. The extension to multi-state games is not meant to be conclusive—it is rather a basis for future work to connect to the well established dynamical models of single-state games.

4.2.1 Multi-state Q-value dynamics

Each player maintains a Q-value function that depends on the state s, the action i, and the stage t. At each iteration, only the selected action is updated based on the perceived reward signal $r_i(t)$. This reward depends on the action i, but may also depend on the state and possibly on the subsequent state after stochastic transition. For the sake of generality, notation for such additional dependencies is omitted but implied:

$$\Delta Q_i(s_t, t) = \begin{cases} \alpha \left(r_i(t) + \gamma \max_j Q_j(s_{t+1}, t) - Q_i(s_t, t) \right) & \text{if } i \text{ selected} \\ & 0 \text{ otherwise.} \end{cases}$$

At any time t, the expected Q-value change for Q-learning in stochastic games can be computed similarly to the single-state derivations given in Section 3.1. Let $p_s = P(s_t = s|t)$ denote the probability for the game to be at state s at time t, and note that updates are only applied to the Q-values of the current game state:

$$\begin{aligned} E\left[\Delta Q_i(s,t)\right] &= E\left[x_i \alpha \left(r_i(t) + \gamma \max_j Q_j(s_{t+1}, t) - Q_i(s_t, t) \right)\right] \\ &= p_s x_i \alpha \left(E\left[r_i(t)\right] + E\left[\gamma \max_j Q_j(s_{t+1}, t)\right] - Q_i(s,t) \right) \\ &= p_s x_i \alpha \left(E\left[r_i(t)\right] + \gamma \sum_{s'} P(s'|s,i) \max_j Q_j(s', t) - Q_i(s,t) \right). \end{aligned}$$

Analogous reasoning leads to the multi-state Q-value dynamics of FAQ-learning:

$$E\left[\Delta Q_i(s,t)\right] = p_s \alpha \left(E\left[r_i(t)\right] + \gamma \sum_{s'} P(s'|s,i) \max_j Q_j(s', t) - Q_i(s,t) \right).$$

Taking the limit of an infinitesimal learning rate for the expected Q-value change under FAQ-learning leads to the following differential form:

$$\frac{dQ_i(s,t)}{dt} = p_s \alpha \left(E\left[r_i(t)\right] + \gamma \sum_{s'} P(s'|s,i) \max_j Q_j(s', t) - Q_i(s,t) \right). \tag{4.3}$$

This equation generalizes the single-state Q-value dynamics that have been derived independently by several authors [Gomes and Kowalczyk, 2009; Tuyls et al., 2003; Wunder et al., 2010]. Specifically, s' and s_t can only take one value s in single-state environments and thus $P(s'|s_t, t) = P(s|s,t) = 1$ and $p_s = P(s_t = s|t) = 1$. By substitution, Equation 4.3 degenerates to the form given in the cited work. Note that these Q-value dynamics hold for several policy-generation schemes, e.g., both for ϵ-greedy and softmax activation. These schemes are encapsulated in the equation by p_s and $P(s'|s,i)$, both of which are dependent on the policy.

4.2.2 Multi-state policy dynamics

Consider FAQ-learning with the softmax activation function to generate policies from Q-values. Recall the softmax activation function:

$$x_i(Q, \tau) = \frac{e^{\tau^{-1} Q_i}}{\sum_j e^{\tau^{-1} Q_j}}.$$

Note that x depends on the state, but notation for that dependence has been dropped for the sake of readability. The derivative of this function has been given in Equation 4.1, and is repeated here for the reader's convenience:

$$\frac{dx_i}{dt} = x_i \tau^{-1} \left[\frac{dQ_i}{dt} - \sum_j x_j \frac{dQ_j}{dt} - \frac{d\tau}{dt} \left(\log x_i - x \log x^T \right) \right].$$

The multi-state Q-value dynamics that have been derived in the previous section can be substituted into this equation to compute the expected policy change for FAQ-learning in stochastic games:

$$\begin{aligned}\frac{dx_i}{dt} = \quad & p_s x_i \tau^{-1} \alpha \left[E\left[r_i(t)\right] - \sum_j x_j E\left[r_j(t)\right] \right] \\ & + p_s x_i \tau^{-1} \alpha \gamma \sum_j x_j \sum_{s'} \Big(P(s'|s,i) - P(s'|s,j) \Big) \max_k Q_k(s',t) \\ & - x_i \left(p_s \alpha + \tau^{-1} \frac{d\tau}{dt} \right) \left(\log x_i - x \log x^T \right). \qquad (4.4)\end{aligned}$$

In the single-state case, $P(s'|s,i)$—$P(s'|s,j) = 0$ and $p_s = 1$, hence this equation degenerates to the FAQ-learning dynamics with varying temperature as derived in Section 4.1. Additionally assuming a constant temperature, the equations further degenerate to the idealized model [Tuyls et al., 2006].

In contrast to the single-state dynamics, the dependency on Q-values does not drop in the policy model. As a result, the learning process is simply higher-dimensional and more complex than can be expressed purely in terms of policy dimensions. This irreducibility and high dimensionality poses additional challenges for the analysis of multi-state dynamics, e.g., even a two-agent two-action two-state problem yields an 8-dimensional phase space, which is hard to visualize.

4.2.3 Dynamics of SARSA

The dynamics of Q-learning and State-Action-Reward-State-Action (SARSA) are equivalent in single-state games, but they do differ in stochastic games. Consider the SARSA update, which is modified from that of Q-learning. Let j be the action selected at time $t+1$:

$$\Delta Q_i(s_t,t) = \begin{cases} \alpha \left(r_i(t) + \gamma Q_j(s_{t+1},t) - Q_i(s_t,t) \right) \text{ if } i \text{ selected} \\ \hfill 0 \text{ otherwise.} \end{cases}$$

Instead of using the max operator as in Q-learning, the SARSA update is based on the Q-value of the selected action in the subsequent state. Analogous to Q-learning, the expected Q-value update becomes:

$$E\left[\Delta Q_i(s,t)\right] = x_i \alpha \left(E\left[r_i(t)\right] + \gamma \sum_{s'} P(s'|s,i) \sum_k x_k(s') Q_k(s',t) - Q_i(s,t) \right).$$

Inspired by FAQ-learning, SARSA can be modified to Frequency Adjusted SARSA (FA-SARSA) by adjusting the update rule:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i(t)}\alpha\Big(r_i(t) + \gamma Q_j(s_{t+1}, t) - Q_i(t)\Big).$$

Taking the infinitesimal limit of this update rule and substituting the resulting Q-value dynamics into the softmax derivate leads to the following dynamics of FA-SARSA:

$$\begin{aligned}\frac{dx_i}{dt} = \quad & p_s x_i \tau^{-1}\alpha\left[E\left[r_i(t)\right] - \sum_j x_j E\left[r_j(t)\right]\right] \\ & + p_s x_i \tau^{-1}\alpha\gamma \sum_j x_j \sum_{s'}\Big(P(s'|s,i) - P(s'|s,j)\Big)\sum_h x_h(s')Q_h(s',t) \\ & - x_i\left(p_s\alpha + \tau^{-1}\frac{d\tau}{dt}\right)\left(\log x_i - x\log x^T\right).\end{aligned}$$

Just like Equation 4.4, this equation degenerates to the previously derived FAQ-learning dynamics in single-state games, where $P(s'|s,i) - P(s'|s,j) = 0$ and $p_s = 1$. This derivation formally proves the equivalence of FAQ-learning and FA-SARSA in single-state games. The specific extended models derived in this and the previous section can be used to study either algorithm in stochastic games.

4.2.4 Networks of learning algorithms

This dissertation models existing learning algorithms and derives dynamical systems to model their dynamics. The challenges going hand-in-hand with the complexity of such multi-state dynamics have inspired a different approach giving rise to less complex multi-state dynamics [Hennes et al., 2010; Vrancx et al., 2008].

The core idea is to adapt learning algorithms from single-state games to stochastic games by feeding them a modified reward signal. To this end, a network of learning algorithms has been conceived [Vrancx et al., 2008]. An agent associates a dedicated learning algorithm, e.g., a Learning Automaton (LA), to each state of the game and control is passed on from one learner to another. Each learner tries to optimize the policy in its state using the standard update rule for single-state games. Only a single learner is active and selects an action at each stage of the game. However, the immediate reward from the environment is not directly fed back to this learner. Instead, when the learner becomes active again, i.e., next time the same state is played, it is informed about the cumulative reward gathered since the last activation and the time that has passed by. The reward feedback $f(t)$ for an agent's automaton LA(s) associated with state s is defined as

$$f(t) = \frac{\Delta r}{\Delta t} = \frac{\sum_{k=t_0(s)}^{t-1} r(k)}{t - t_0(s)}\ , \tag{4.5}$$

where $r(k)$ is the immediate reward for the agent in epoch k and $t_0(s)$ is the last occurrence function that determines when states s was visited last. The reward feedback in epoch t equals the cumulative reward Δr divided by time-frame Δt. The cumulative reward Δr is the sum over all immediate rewards gathered in all states beginning with epoch $t_0(s)$ and including the last epoch $t-1$. The time-frame Δt measures the number of epochs that have passed since automaton LA(s) has been active last. This definition means the state policy is updated using the average stage reward over the interim immediate rewards. Initial experiments solely consider learning automata [Vrancx et al., 2008]. However, the method is more general: any learning algorithm that is applicable to single-state games can be aggregated in a network, which is then applicable to stochastic games.

Learning automata do not exhibit any exploration, a fact that may lead to convergence to local optima or even non-convergence in multi-agent learning [Hennes et al., 2010]. As a resolution, this concept of networks of learning algorithms has been extended to incorporate exploration in *Reverse Engineering State-coupled replicator dynamics injected with the Q-learning Boltzmann mutation scheme* (RESQ-learning) [Hennes et al., 2010]. The designed learning algorithm inherits the convergence behavior of the reverse engineered dynamical system. In particular. RESQ-learning converges to pure as well as mixed Nash equilibria in a selection of stateless and stochastic multi-agent games.

In related work, the idea of a network of learning algorithms has been combined with *optimism in uncertainty*, inspired by the algorithm R-max [Brafman and Tennenholtz, 2002], to create the algorithmic framework PEPPER that can transform any single-state learner into a learner for stochastic games [Crandall, 2012]. However, this framework assumes agents can observe the joint action, and the empirical investigation is lacking an analytical underpinning.

4.3 Lenient learning in cooperative games

Recently, an evolutionary model of Lenient Q-learning (LQ) has been proposed, providing theoretical guarantees of convergence to the global optimum in cooperative multi-agent learning [Panait et al., 2008]. However, experiments reveal discrepancies between the dynamics of the evolutionary model and the actual learning behavior of the Lenient Q-learning algorithm, which undermines its theoretical foundation. Moreover it turns out that the predicted behavior of the model is more desirable than the observed behavior of the algorithm. The variant Lenient Frequency Adjusted Q-learning (LFAQ) combines the advantages of lenient learning in coordination games with FAQ-learning, inheriting the theoretical guarantees and resolving this issue [Bloembergen et al., 2010a,b].

The advantages of LFAQ are demonstrated by comparing the evolutionary dynamics of lenient vs non-lenient Frequency Adjusted Q-learning. In addition, the behavior, convergence properties and performance of these two learning algorithms is analyzed empirically. The algorithms are evaluated in the Battle of the Sexes (BoS) and the Stag Hunt (SH) games with the following payoff matrices, while compensating for intrinsic learning speed differences.

$$\begin{array}{c} \begin{array}{cc} & \begin{array}{cc} \mathsf{O} & \mathsf{F} \end{array} \\ \begin{array}{c} \mathsf{O} \\ \mathsf{F} \end{array} & \left(\begin{array}{cc} 1, \frac{1}{2} & 0, 0 \\ 0, 0 & \frac{1}{2}, 1 \end{array}\right) \end{array} \\ \textbf{Battle of the Sexes} \end{array} \qquad \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} \mathsf{S} & \mathsf{H} \end{array} \\ \begin{array}{c} \mathsf{S} \\ \mathsf{H} \end{array} & \left(\begin{array}{cc} 1, 1 & 0, \frac{2}{3} \\ \frac{2}{3}, 0 & \frac{2}{3}, \frac{2}{3} \end{array}\right) \end{array} \\ \textbf{Stag Hunt} \end{array}$$

Significant deviations arise from the introduction of leniency, leading to profound performance gains in coordination games against both lenient and non-lenient learners.

4.3.1 Lenient Frequency Adjusted Q-learning

It has been shown that leniency, i.e., forgiving initial mis-coordination, can greatly improve the accuracy of an agent's reward estimation in the beginning of the cooperative learning process [Panait et al., 2008]. It thereby overcomes the problem that initial mis-coordination may lead learners to get stuck in local optima with mediocre payoffs. Leniency thus increases the probability of reaching the global optimum. Leniency towards others can be achieved by having the agent collect κ rewards for a single action before updating the value of this action based on the highest of those κ rewards [Panait et al., 2008].

An evolutionary model of LQ delivers formal convergence guarantees based on the idealized model of Q-learning, which has been derived under the assumption that all actions are updated equally quickly [Tuyls et al., 2003]. However, the action-values in Q-learning are updated asynchronously and thus at different frequencies: the value of an action is only updated when the action is selected. Chapter 3 has shown that the idealized evolutionary model describes more rational behavior than the Q-learning algorithm actually exhibits. Consequently, the variation Frequency Adjusted Q-learning (FAQ) has been introduced, which weights the action-value update inversely proportionally to the action-selection probability, thereby removing initialization dependencies:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i}\alpha \left[r(t+1) + \gamma \max_j Q_j(t) - Q_i(t) \right].$$

The variation Lenient Frequency Adjusted Q-learning (LFAQ) combines the improvements of FAQ and Lenient Q-learning. The action-value update rule of LFAQ is equal to that of FAQ; the difference is that the lenient version collects κ rewards before updating its Q-values based on the highest of those rewards. An elaborate explanation of this algorithm can be found in published work [Bloembergen et al., 2010b].

4.3.2 Experiments and results

This section provides a validation of the proposed LFAQ algorithm, as well as an empirical comparison to non-lenient FAQ. A more elaborate evaluation of the performance of lenient vs. non-lenient learning algorithms can be found elsewhere [Bloembergen et al., 2010a].

Figure 4.3 presents an overview of the behavior of Lenient Q-learning and Lenient FAQ-learning in the Stag Hunt game. Similarly to the validation of FAQ-learning in

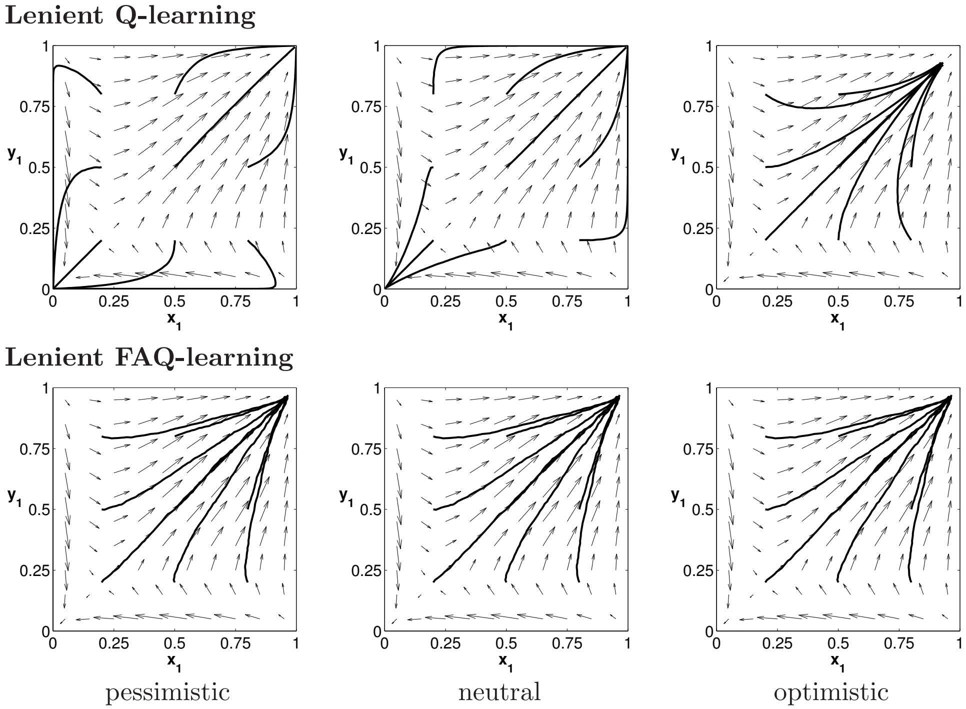


Figure 4.3: Trajectories of LQ-learning and LFAQ-learning (solid lines), and the LFAQ dynamics (arrows) in the Stag Hunt.

Section 3.2, the figure shows different initialization settings for the Q-values: pessimistic (left), neutral (center) and optimistic (right). The arrows represent the directional field plot of the lenient evolutionary model; the lines follow learning traces of the algorithm. These results show that the behavior of LQ deviates considerably from the evolutionary model, and depends on the initialization. LFAQ on the other hand is coherent across different initialization values, and follows the evolutionary model precisely. Moreover, the basin of attraction for the global optimum $(1, 1)$ is larger for LFAQ than for LQ, resulting in an overall payoff advantage for LFAQ.

Figure 4.4 shows policy trajectories of FAQ, LFAQ, and one versus the other in Battle of the Sexes (BoS) and the Stag Hunt (SH). In BoS, LFAQ has an advantage against non-lenient FAQ when the two are in competition, indicated by a larger basin of attraction for its preferred equilibrium at $(0, 0)$. In that equilibrium, LFAQ receives twice as much reward as FAQ. Intuitively, the optimism of leniency makes LFAQ insist on its preferred equilibrium just a bit more such that FAQ adapts toward it if undecided, i.e., x close to 0.5. In the SH, LFAQ outperforms FAQ in self-play with a larger basin of attraction for the global optimum at $(1, 1)$. Against each other, both algorithms converge to the same payoff.

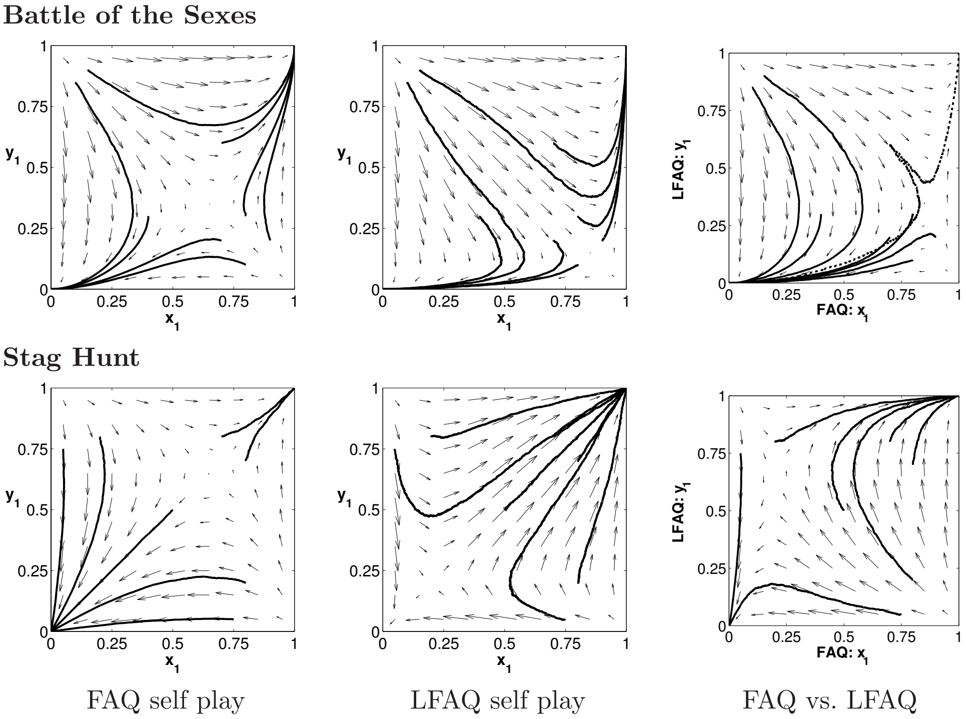


Figure 4.4: Comparing lenient and non-lenient FAQ-learning in two cooperative games.

Finally, Figure 4.5 shows the average reward over time for FAQ (solid) and LFAQ (dotted) in self-play, as well as for FAQ (dashed) versus LFAQ (dash-dot). Making explicit what could be conjectured from the behavioral analysis, LFAQ is advantageous to FAQ, by virtue of achieving either a higher or similar average reward.

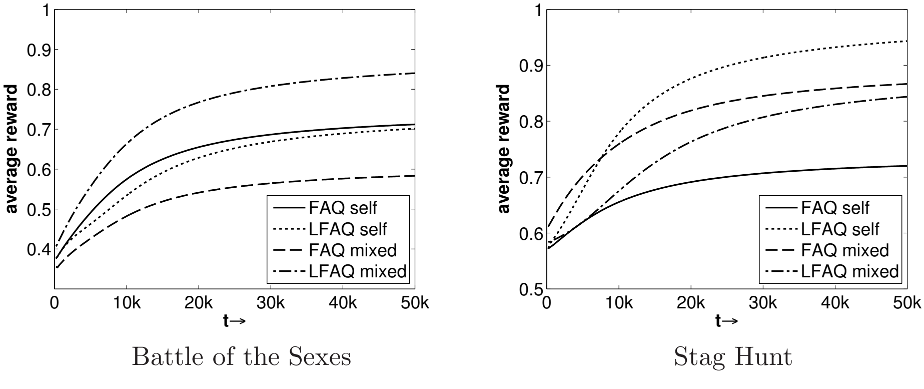


Figure 4.5: Average reward plots for LFAQ-learning and FAQ-learning in self-play and against each other.

4.3.3 Discussion

The proposed LFAQ algorithm combines insights from FAQ-learning, as presented in Chapter 3, and LQ-learning [Panait et al., 2008] and inherits the theoretical advantages of both. Empirical comparisons confirm that the LFAQ algorithm is consistent with the evolutionary model [Panait et al., 2008], whereas the LQ algorithm may deviate considerably. Furthermore, the behavior of LFAQ is independent of the initialization of the Q-values. In general, LFAQ performs at least as well as non-lenient learning in coordination games. As such, leniency is the preferable and safe choice in cooperative multi-agent learning.

4.4 Summary

This chapter has extended the framework for multi-agent reinforcement learning in three ways: First, the dynamics of FAQ-learning have been derived for exploration rates that may vary over time. This model has been used to design an appropriate temperature function that increases the probability of converging to the global optimum. Second, the Q-value and policy dynamics have been extended to stochastic games with multiple states. Dynamics of FAQ-learning and SARSA have been derived and a related approach based on networks of learning algorithms has been outlined. Third, Lenient FAQ-learning is proposed to increase the convergence to global optima in cooperative games. Experiments in simple games have illustrated the theoretical findings.

5 New perspectives

This chapter presents two new perspectives on multi-agent learning dynamics. Even using the framework described in the previous chapters, understanding learning under conditions that vary over time remains a challenging task. To aid in this challenge, the next section introduces an *orthogonal* visualization that makes it possible to gain better insight into time-dependent properties of multi-agent learning dynamics [Kaisers, 2009]. Section 5.1 demonstrates how this tool facilitates designing time-dependent parameters. Subsequently, a second new perspective reveals the common ground of reinforcement-learning algorithms and gradient ascent. From this view it becomes apparent that multi-agent reinforcement learning implements on-policy stochastic gradient ascent on the payoff function [Kaisers et al., 2012; Kaisers and Tuyls, 2012]. This result fills a gap between gradient ascent and evolutionary game theory, which have evolved as separate streams of inquiry and are now united in the same framework. Given the close relationship established here, insights in convergent gradient ascent dynamics can, for example, be taken as an inspiration for analogous independent reinforcement-learning improvements.

5.1 An orthogonal visualization of learning dynamics

This section introduces a new perspective on the reinforcement-learning process described by the replicator dynamics, providing a tool for designing time-dependent parameters of the game or the learning process. The learning dynamics are commonly visualized by showing the directional field plot of the replicator dynamics or showing policy trajectories with the time dimension collapsed into a surface. Both views work well for dynamics that do not change over time but provide little guidance when the game or the learning algorithm uses a parameter that is time-dependent. In particular, the directional field plot can only capture the dynamics at one point in time. Hence, several independent plots are needed for changing dynamics and a gap remains in the transition between them. The trajectory view becomes unclear when cycles occur or the dynamics change, in which case lines may intersect and clutter the plot. Furthermore, reducing the time dimension into a flat surface hinders the interpretation of time-dependent artifacts. In addition, the higher the resolution (the more trajectories that are plotted), the more crowded the plot and the harder it becomes to interpret. As a result, parameter tuning is a cumbersome task that often results in ad hoc trial and error approaches. To tackle these problems, a new perspective is proposed that elicits more information from dynamical systems, especially for time-dependent dynamics, with the goal of facilitating the systematic design of time-dependent parameters.

5.1.1 Method

This section shows the learning process in a new perspective, which is orthogonal to viewing policy trajectories in the classical way. Trajectories have a policy component for each player and a time dimension. Figure 5.1 shows 20 trajectories from three perspectives: (1) the classical view as a phase space diagram in the policy space with the time dimension collapsed, (2) an expanded view showing both the policy space and the time dimension, and (3) the newly proposed orthogonal view. Instead of looking at it from the top down, one can cut slices at different points in time and look at the distribution of trajectory points where they intersect these slices that are orthogonal to the top-down view. Each slice shows the density of trajectory points at a certain time. Considering distributions rather than single trajectories provides a more holistic view of the learning process. In the end, learning is a homeomorphic time-dependent

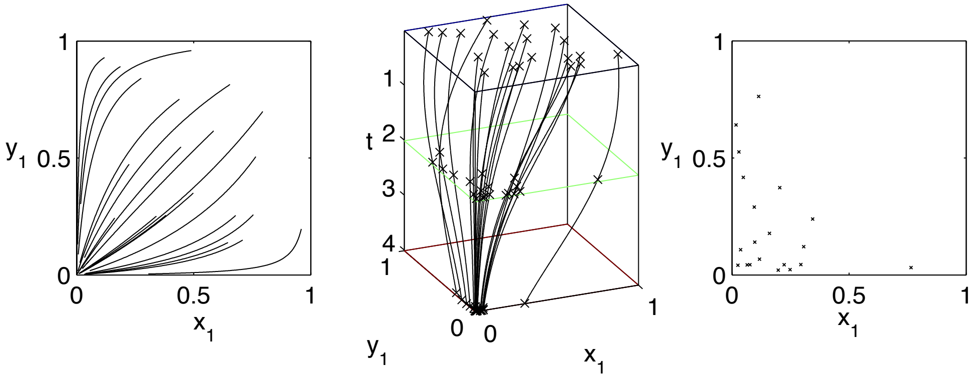


Figure 5.1: An expanded view of 20 policy trajectories (middle), the common perspective collapsing the time dimension (left), showing the trajectories as a flat image, and the proposed orthogonal perspective (right), showing the second slice that intersects the trajectories at the indicated points.

transformation of the policy space. As such, its influence on the whole space can be examined, e.g., by looking at the spacing between the trajectories, rather than only looking at individual policy trajectories. To do so, a set of particles is drawn from an initial distribution and subjected to a velocity field defined by the replicator dynamics. Figure 5.1 shows a uniform initial distribution in the top slice of the expanded view. As time evolves, the distribution is transformed and the density of the particles changes, until it is almost converged as in the bottom slice of the expanded view. This diagram makes it possible to make statements of the following kind: assuming any policy was initially equally likely and these policies evolve according to the replicator dynamics, then after time t has passed, p percent of the policies have converged to attractor a with at most distance ϵ.

After some time, the simulation can be stopped and labels can be applied according to the eventual distribution. A certain percentage of particles can be considered converged to some attractors, assuming they are in the neighborhood of a stable point and that point is attracting in that neighborhood. Other particles can be labeled as not converged. Finally, these labels can be applied to earlier slices including the initial slice, revealing the basins of attraction. Although these basins can also be read from the directional field plot of the replicator dynamics, this approach is more general as it can be applied to dynamics that are controlled by a time-dependent parameter.

In addition, this diagram makes possible judging the convergence of a fraction of the policy space that is bound by a surface by considering the velocity field only on that surface. Due to the fact that the dynamics describe a continuous process and the transformation by the replicator dynamics is a homeomorphism, everything that is added or subtracted from the trapped percentage has to go through the surface. This observation is related to the *divergence theorem* from physics [Feynman et al., 2005]. It enables focussing attention on the surface that may be just a small subspace of the whole policy space, e.g., a hypersphere with radius ϵ around an attractor. In many cases, the velocity field in this small neighborhood can be guaranteed to be rather static although the dynamics of other areas of the policy space may change quite substantially. A proof for the convergence of FAQ-learning has been proposed based on this connection to divergence [Kianercy and Galstyan, 2012], complementing the arguments given in Chapter 3.

This approach makes it possible to employ an arbitrary initial distribution, which can be used to model specific prior knowledge about the players' behavior. Commonly, every policy is assumed to be initially equally likely, i.e., applying an initially uniform distribution. Furthermore, the policy distribution can also be generated from Q-value distributions, in case a Q-learning algorithm should be modeled. Using a similar evolution as the replicator dynamics in the Q-value space, the distribution can be evolved, enabling a comparison of Boltzmann exploration to other exploration schemes that do not have a bijective action-selection function[1] and can therefore not be solely described by dynamics in the policy space.

[1] Strictly speaking, Boltzmann action selection is also not a bijection, as it leaves one degree of freedom when computing Q-values from policies. However, each policy change relates to a Q-value change and vice versa, which is not the case in other exploration schemes such as epsilon-greedy.

5.1.2 Experiments

This section demonstrates the proposed methodology on an example game that is controlled by a parameter that may change its value at one point in time. The game describes the following situation:

> There are two new standards that enable communication via different protocols. The consumers and suppliers can be described by probability vectors that show which standard is supported by which fraction of the population. One protocol is 20% more energy efficient, hence the government wants to support that standard. Usually, the profit of the consumers and suppliers are directly proportional to the fraction of the opposite type that supports their standard. However, the government decides to subsidize early adopters of the better protocol.
>
> Such subsidies are expensive and the government only wants to spend as much as necessary. They have no market research information and consider any distribution of supporters on both sides equally likely. Furthermore, they know that the supporters are rational and their fractions will change according to the replicator dynamics. The question is, how long is the subsidy necessary to guarantee that the better standard is adopted in 95% of the possible initial policies.

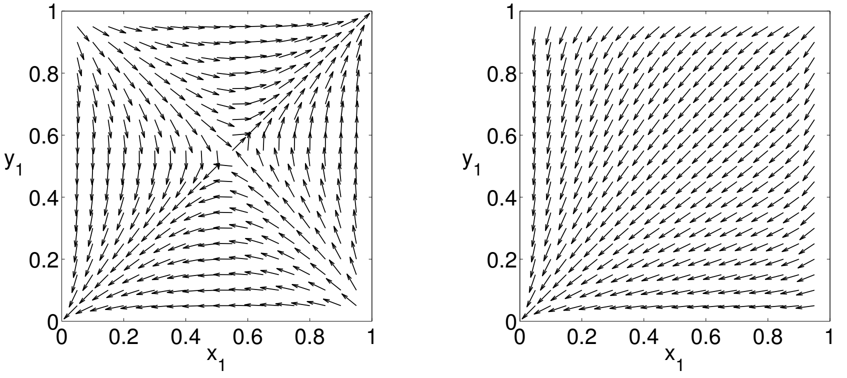
This scenario is a variation of the pure coordination game. A subsidy parameter $s \in \{0, 11\}$ is added, which can be used to make one action dominant. As a result, coordination on the Pareto optimal equilibrium is facilitated. Figure 5.2 displays the payoff bi-matrix game numerically.

The dynamics of the game can be visualized by showing the directional field plot of the replicator dynamics as shown in Figure 5.3. It can be observed that a large fraction of the policy space would converge to the suboptimal standard in the unsubsidized game, while all policies would converge to the optimum in the subsidized game. However, it is difficult to derive the correct time to switch between the two games.

The second classical way to look at the dynamics are policy trajectories. These will follow the directional change and are depicted in Figure 5.4. Similar to the replicator dynamics, this view neatly explains the dynamics of the individual parts of the game, but it is not suitable to infer the right time to switch from the one to the other.

$$\begin{array}{c} \\ \mathrm{S}_1 \\ \mathrm{S}_2 \end{array} \begin{array}{cc} \mathrm{S}_1 & \mathrm{S}_2 \\ \multicolumn{2}{c}{\left(\begin{array}{cc} 10,10 & 0,\mathrm{s} \\ \mathrm{s},0 & 12,12 \end{array}\right)} \end{array} \quad \begin{array}{c} \\ \mathrm{S}_1 \\ \mathrm{S}_2 \end{array} \begin{array}{cc} \mathrm{S}_1 & \mathrm{S}_2 \\ \multicolumn{2}{c}{\left(\begin{array}{cc} 10,10 & 0,0 \\ 0,0 & 12,12 \end{array}\right)} \end{array} \quad \begin{array}{c} \\ \mathrm{S}_1 \\ \mathrm{S}_2 \end{array} \begin{array}{cc} \mathrm{S}_1 & \mathrm{S}_2 \\ \multicolumn{2}{c}{\left(\begin{array}{cc} 10,10 & 0,11 \\ 11,0 & 12,12 \end{array}\right)} \end{array}$$

Figure 5.2: The payoff bi-matrix form of the subsidy game (left) and its realizations for $s = 0$ (middle) and $s = 11$ (right). Player one chooses a row, player two chooses a column. The first number of the selected action combination represents the payoff to player one and the second number the payoff to player two.



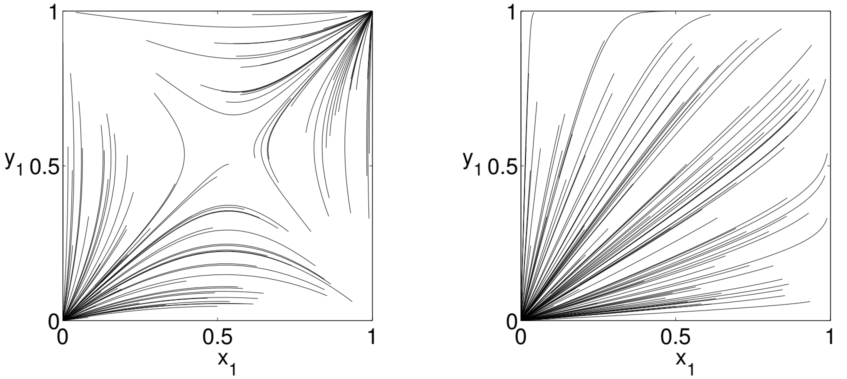
(a) no subsidy

(b) subsidy $s = 11$

Figure 5.3: The learning dynamics of the game with and without subsidy.

Another possible approach is the visualization of trajectories with transitions from one game to the other at different points in time. Figure 5.5 shows the trajectories of the subsidy game when transition from $s = 11$ to $s = 0$ takes place at $t = \{0.1, 0.3, 0.5\}$. Although it can be observed that fewer trajectories converge suboptimally the later the switch occurs, this approach requires guessing the right time of transition. Furthermore, the view is cluttered by intersecting lines and readability does not make it possible to increase the number of trajectories.

To obtain insight into the time-dependent artifacts of these dynamics, the new visualization will be applied. Answering the question of when to switch requires 2 steps:



(a) no subsidy

(b) subsidy $s = 11$

Figure 5.4: Trajectories with a length of 4 units of continuous time in the game without subsidy (left) and with subsidy (right).

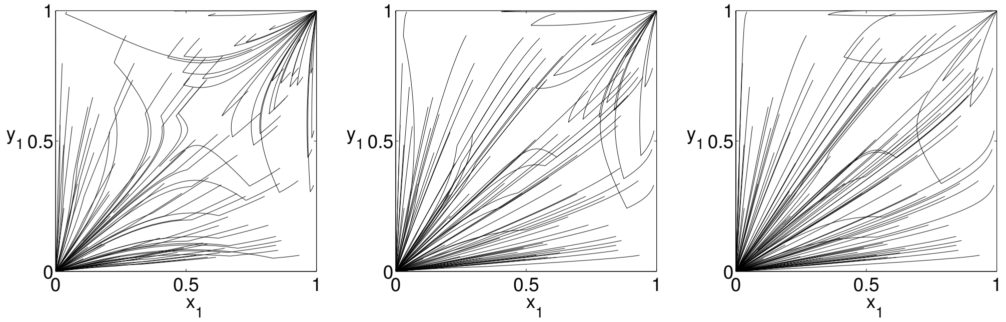


Figure 5.5: The trajectory plot for the subsidy game with transition from the subsidized to the unsubsidized game at $t = \{0.1, 0.3, 0.5\}$ (left to right).

- Determine the part of the policy space for which trajectories converge optimally in the unsubsidized game.
- Determine the time when the subsidized dynamics have driven 95% of the initial policies into the previously determined subspace.

Step one is shown in Figure 5.6. Particles are drawn from a uniform initial distribution and evolved according to the replicator dynamics. After $t = 1.2$, the particles are considered converged and receive a label. Subsequently, the label is applied to all slices before plotting. From the labels on the initial slice, the basin boundary is deduced using a linear best fit, which is marked by the dashed line.

In step 2, shown in Figure 5.7, the boundary that has been inferred from step one is used to monitor the percentage of the initial policy space that would converge to the optimum if the game was switched at that time instance. The simulation advances until

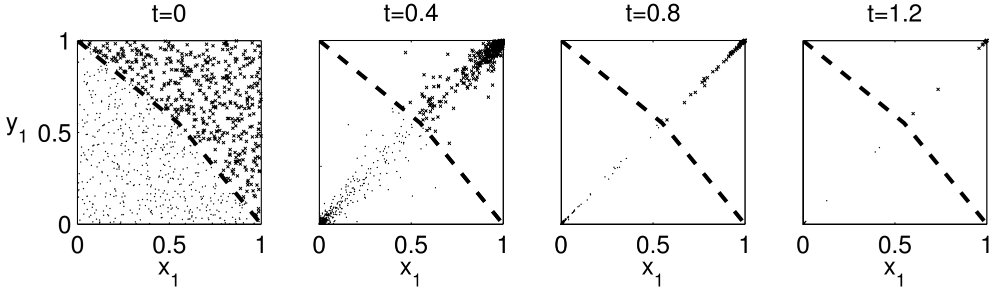


Figure 5.6: This figure shows the evolution of particles drawn from a uniform initial distribution, revealing the basins of attraction of the unsubsidized game. Labels are applied according to the last slice and the dashed line is inferred from the labels to be the boundary between the basins of attraction.

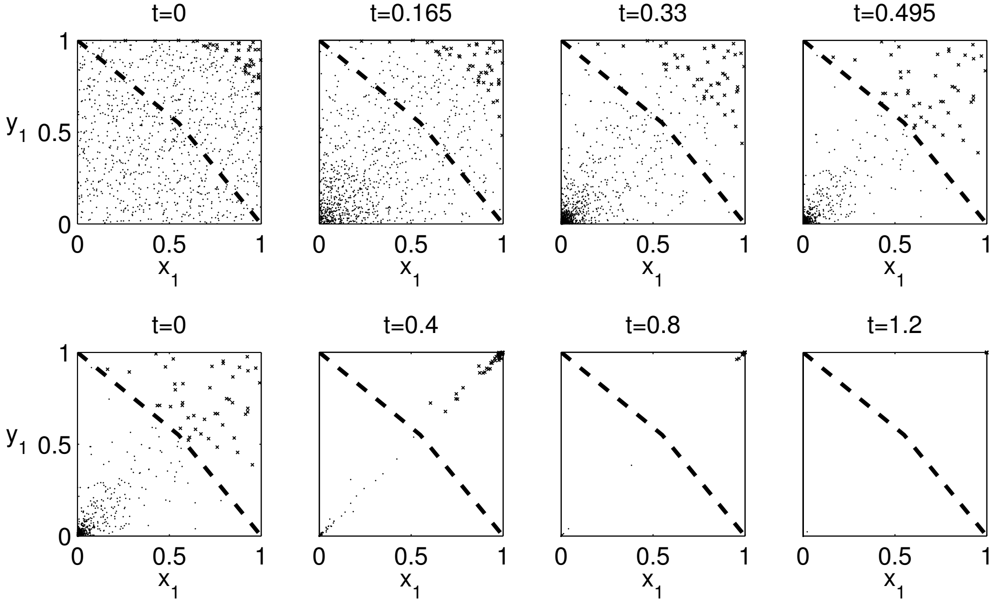


Figure 5.7: The top row shows the evolution in the subsidized game until 95% of the policy space are in the basin for the global optimum of the unsubsidized game. The lower row shows the further evolution in the unsubsidized game.

the subsidized dynamics have pushed 95% of the initial policies into the basin of attraction of the global optimum in the unsubsidized game. Then, the game is switched and the simulation shows convergence to the respective attractors. Repeating the experiment $n = 1000$ times, the time to bring 95% to the basin is found to be 0.495 ± 0.0357 (indicating one standard deviation). A histogram of the distribution of convergence times for this experiment is given in Figure 5.8.

5.1.3 Discussion

As demonstrated on the subsidy game, the orthogonal visualization allows the systematic study and design of time-dependent parameters to achieve a specific convergence behavior. The parameter-design methodology can be transferred to other parameters that change the replicator dynamics, most prominently the temperature function for Q-learning with a Boltzmann exploration scheme. Choosing an appropriate temperature function has long been approached in an ad hoc manner and can now be tackled systematically to achieve a desired convergence distribution. While a rather simple example game was studied for the sake of clarity, the approach is general in the number of actions and can be applied to arbitrary initial distributions. In addition, it naturally generalizes to any number of agents.

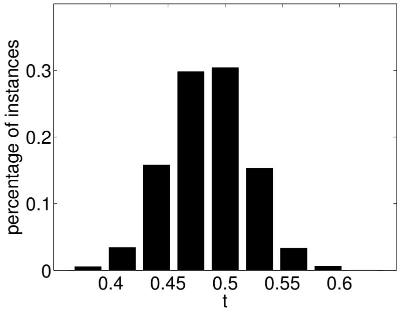


Figure 5.8: Histogram of times at which the velocity field of the subsidized game has driven 95% of the particles into the basin of attraction of the global optimum in the unsubsidized game. The sample size is $n = 1000$, with a mean of 0.495 and a standard deviation of 0.0357.

The ideas presented in this section have a lot of potential to be developed further. The current approach can be seen as a particle simulation, where the replicator dynamics determine the velocity field that describes the movement of each particle, and the particle density describes a probability distribution. This probability distribution can equivalently be described as a continuous probability density function, deriving the density change directly from the replicator dynamics. Such a functional model removes the stochasticity introduced by approximating the probability density by quantized particles. The concept of density and change of density has been related to divergence of a vector field, primarily used in physics, and supports an auxiliary proof to the one given in Section 3.3 for the convergence of FAQ-learning [Crandall et al., 2011]. A similar analysis may be transferred to the distribution of Q-values. Doing so enables the comparison of exploration schemes such as Boltzmann and epsilon-greedy exploration. Finally, this model is extendable to multiple states and continuous strategy spaces, which will compliment the theoretical framework for multi-agent learning.

5.2 Reinforcement learning as stochastic gradient ascent

Gradient ascent on the expected reward has been used to derive convergence guarantees in two-player two-action games, at the expense of strong assumptions such as full information about the game being available to the players. In contrast, independent multi-agent reinforcement learning requires less information: it uses feedback from discrete interactions with the environment instead. Algorithms such as *Cross learning*, variations of *Q-learning* and *Regret minimization* have been related to the replicator dynamics from evolutionary game theory to obtain insight into their convergence behavior. A formal analysis reveals that these algorithms implement on-policy stochastic gradient ascent, which bridges the gap between two streams of research. Cross learning for example exhibits convergence behavior equivalent to *Infinitesimal Gradient Ascent.*

In addition to the derivations, directional field plots of the learning dynamics in representative classes of two-player two-action games illustrate the similarities and strengthen the theoretical findings.

The remainder of this section contains an overview of the dynamics of the different algorithms, and highlights their similarities. First, the evolutionary game theoretic models that have been derived for Cross learning, Frequency Adjusted Q-learning and regret minimization are compared using a unified notation. Next, the similarities between these evolutionary dynamics and the gradient ascent algorithms are derived for two-player two-action games. Finally, these findings are generalized to normal-form games.

5.2.1 Evolutionary dynamics of reinforcement learning

Independent reinforcement learning starts from a different premise than gradient ascent. Instead of assuming full knowledge of the value function, a reinforcement-learning agent learns from scratch by repeatedly interacting with its environment. After taking an action, the agent perceives the resulting state of the environment and receives a reward that captures the desirability of that state and the cost of the action. While the single-agent reinforcement-learning problem is well-defined as a Markov decision process, the multi-agent case is more complex. As state transitions and rewards are influenced by the joint action of all agents, the Markov property is no longer satisfied from a single agents' point of view. In essence, each agent is chasing its optimal policy, which depends on what the other agents do—and since they change as well, all agents chase a moving target. Nevertheless, single-agent reinforcement-learning algorithms have been shown to produce good results in the multi-agent case [Busoniu et al., 2008]. Three independent reinforcement algorithms are examined in detail here: the policy iterator Cross learning, and the value iterators regret minimization and Q-learning. These algorithms have been introduced in Chapter 2 but are revisited here for the reader's convenience.

Cross learning (CL) was the first algorithm to be linked to a dynamical system from evolutionary game theory [Börgers and Sarin, 1997]. As described in Section 2.5.1, the learning dynamics of CL in the limit of an infinitesimal update step approach the replicator dynamics of Equation 2.6. The link between a policy learner like CL and a dynamical system in the policy space may be rather straight-forward. However, the link has been extended to value-based learners as well. A model of Q-learning with the Boltzman update scheme has been proposed [Tuyls et al., 2006], given the additional assumption of updating all actions simultaneously. The variation Frequency-Adjusted Q-learning (FAQ-learning), discussed in detail in Chapter 3, implements this model by modulating the update rule inversely proportionally to the action probability x_i, thereby approximating simultaneous action updates:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i}\alpha\left[r_i(t) - Q_i(t)\right].$$

As derived in Chapter 3, the dynamical system that corresponds to this update rule can be decomposed into terms for exploitation (the replicator dynamics) and exploration (randomization based on the Boltzmann mechanism), revealing its relation to the replicator dynamics:

$$\dot{x}_i = \frac{\alpha x_i}{\tau} \underbrace{\left[e_i A y^T - x A y^T\right]}_{\text{exploitation}} - \alpha x_i \underbrace{\left[\log x_i - \sum_k x_k \log x_k\right]}_{\text{exploration}}.$$

Recently, the evolutionary framework has also been extended to the Polynomial Weights algorithm, which implements regret minimization [Klos et al., 2010]. Despite the great difference in update rule and policy generation (see Eq. 2.3), the infinitesimal limit has been linked to a dynamical system with CL dynamics in the numerator.

$$\dot{x}_i = \frac{\alpha x_i \left[e_i A y^T - x A y^T\right]}{1 - \alpha \left[\max_k e_k A y^T - x A y^T\right]}.$$

The denominator scales the learning-rate proportional to the best action's update magnitude.

5.2.2 Similarities in two-player two-action games

For two-agent two-action games, the dynamics can be simplified. Let $h = (1, -1)$, $x = (x_1, 1 - x_1)$ and $y = (y_1, 1 - y_1)$. The dynamics are completely described by the pair $(\dot{x}_1, \dot{y}_1)$, which denote the probability changes of the first actions for both players. For CL in self-play, this leads to the following simplified form:

$$\begin{aligned} \dot{x}_1 &= x_1(1 - x_1)\left[y_1 h A h^T + A_{12} - A_{22}\right] \\ \dot{y}_1 &= y_1(1 - y_1)\left[x_1 h B h^T + B_{21} - B_{22}\right]. \end{aligned}$$

The second player's update $\dot{y}_1$ is completely analogous to $\dot{x}_1$, and will be omitted in the subsequent discussion. To simplify the notation for two-action games, let $\eth = e_1 A y^T - e_2 A y^T = y_1 h A h^T + A_{12} - A_{22}$ denote the gradient. The simplified FAQ-learning dynamics read

$$\dot{x}_1 = \alpha x_1(1 - x_1)\left[\frac{\eth}{\tau} - \log\left(\frac{x_1}{1 - x_1}\right)\right].$$

The dynamics of Regret Minimization (RM) are slightly more complex, as the denominator depends on which action gives the highest reward. This information can be derived from the gradient: the expected reward for the first action will be a maximum iff $\eth \geqslant 0$. Using this insight, the dynamics of RM in two-action games can be written as follows:

$$\dot{x}_1 = \alpha x_1(1 - x_1)\eth \cdot \begin{cases} (1 + \alpha x_1 \eth)^{-1} & \text{if } \eth < 0 \\ (1 - \alpha(1 - x_1)\eth)^{-1} & \text{otherwise.} \end{cases}$$

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 $\eth$ of the expected reward V, which in two-player two-action games can be written in the following form:

$$\begin{aligned}\frac{\partial V(x,y)}{\partial x_1} &= \frac{\partial}{\partial x_1}(x_1, 1-x_1)A\begin{pmatrix} y_1 \\ 1-y_1 \end{pmatrix}\\ &= y_1(A_{11}-A_{12}-A_{21}+A_{22})+A_{12}-A_{22}\\ &= y_1 hAh^T + A_{12}-A_{22}\\ &= \eth.\end{aligned}$$

This derivation reduces the dynamics of the update rule for IGA in two-player two-action games to $\dot{x_1} = \alpha\eth$.

The extension of the dynamics of IGA to IGA-WoLF and WPL are straightforward (see Section 2.5.4). Table 5.1 lists the dynamics of the six discussed algorithms: IGA [Singh et al., 2000], WoLF [Bowling and Veloso, 2002], WPL [Abdallah and Lesser, 2008], CL [Börgers and Sarin, 1997], FAQ [Kaisers and Tuyls, 2010] and RM [Klos et al., 2010]. It is immediately clear from this table that *all algorithms have the same basic term in their dynamics*: the gradient $\eth$. Depending on the algorithm, the gradient is scaled with a learning-speed modulation. FAQ-learning yields the only dynamics that additionally add exploration terms to the process.

Next, the learning dynamics are juxtaposed in representative two-player two-action games. Three distinct classes can be identified [Gintis, 2009]: games with one pure Nash equilibrium (e.g. Prisoners' Dilemma); games with two pure and one mixed NE (e.g. Battle of the Sexes); and games with one mixed NE (e.g. Matching Pennies). The normalized payoff bi-matrices of these games are as presented in Figure 5.9.

Since the joint policies in two-player two-action games are completely defined by the pair (x_1, y_1), it is possible to plot the learning process in the unit square. Trajectories can be drawn by following the learning from a specific initial joint policy. Figure 5.10 illustrates learning trajectories in the Matching Pennies, where IGA and CL both cycle

Table 5.1: This table shows an overview of the learning dynamics, rewritten for the specific case of two-agent two-action games. For simplicity, the common gradient is abbreviated $\eth = y_1 hAh^T + A_{12} - A_{22}$.

Alg.	Evolutionary model $\dot{x}_1$	Type
IGA	$\alpha\eth$	gradient ascent
WoLF	$\eth \cdot \begin{cases} \alpha_{min} & \text{if } V(x,y) > V(x^e,y) \\ \alpha_{max} & \text{otherwise} \end{cases}$	
WPL	$\alpha\eth \cdot \begin{cases} x_1 & \text{if } \eth < 0 \\ (1-x_1) & \text{otherwise} \end{cases}$	
CL	$\alpha x_1(1-x_1)\ \eth$	replicator dynamics
FAQ	$\alpha x_1(1-x_1)\left[\eth \cdot \tau^{-1} - \log\left(\frac{x_1}{1-x_1}\right)\right]$	
RM	$\alpha x_1(1-x_1)\ \eth \cdot \begin{cases} (1+\alpha x_1 \eth)^{-1} & \text{if } \eth < 0 \\ (1-\alpha(1-x_1)\eth)^{-1} & \text{otherwise} \end{cases}$	

$$\begin{array}{ccc} & \mathsf{C} & \mathsf{D} \\ \mathsf{C} & \frac{3}{5},\frac{3}{5} & 0,1 \\ \mathsf{D} & 1,0 & \frac{1}{5},\frac{1}{5} \end{array} \qquad \begin{array}{ccc} & \mathsf{O} & \mathsf{F} \\ \mathsf{O} & 1,\frac{1}{2} & 0,0 \\ \mathsf{F} & 0,0 & \frac{1}{2},1 \end{array} \qquad \begin{array}{ccc} & \mathsf{H} & \mathsf{T} \\ \mathsf{H} & 1,0 & 0,1 \\ \mathsf{T} & 0,1 & 1,0 \end{array}$$

Prisoners' Dilemma — Battle of the Sexes — Matching Pennies

Figure 5.9: Normalized payoff matrices for three representative two-player two-action games.

around the Nash equilibrium. RM is omitted since it is indistinguishable from CL. The other three algorithms (WoLF, WPL, FAQ) spiral inwards and eventually converge, but do so in a different manner. The dynamics of WoLF clearly show the effect of the *Win or Learn Fast* scheme, switching between the two distinct learning step values at $x_1 = 0.5$ and $y_1 = 0.5$. Similarly, the orbits of WPL yield a less steep corner when switching between the two update schemes.

Figure 5.11 shows a directional field plot of the learning dynamics in the Prisoner's Dilemma, Battle of the Sexes, and Matching Pennies game. Each arrow indicates the direction of change at that point (x_1, y_1) in the policy space. Again, the dynamics of RM can be considered equivalent to CL. The figure illustrates the high similarity between all algorithms in the first two games. They all share the same convergence properties, and follow similar trajectories. The dynamics of IGA and WoLF in the Prisoners' Dilemma show the need for the $\mathtt{projection}$ function to prevent the update from taking the policies x and y out of the valid policy space. The largest variety is observed in the MP game as illustrated before in Figure 5.10.

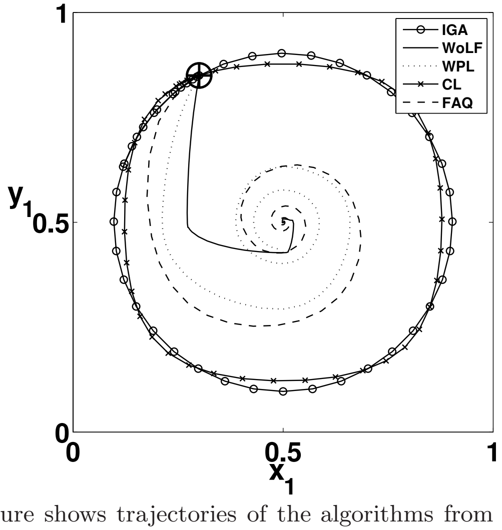


Figure 5.10: This figure shows trajectories of the algorithms from the same starting point (indicated with ⊕) in the Matching Pennies game. IGA and CL yield stable cycles, while WoLF, WPL and FAQ-learning converge to the mixed Nash equilibrium $\left(\frac{1}{2}, \frac{1}{2}\right)$. WoLF uses $\frac{\alpha_{min}}{\alpha_{max}} = 0.1$, and FAQ uses $\tau = 0.1$.

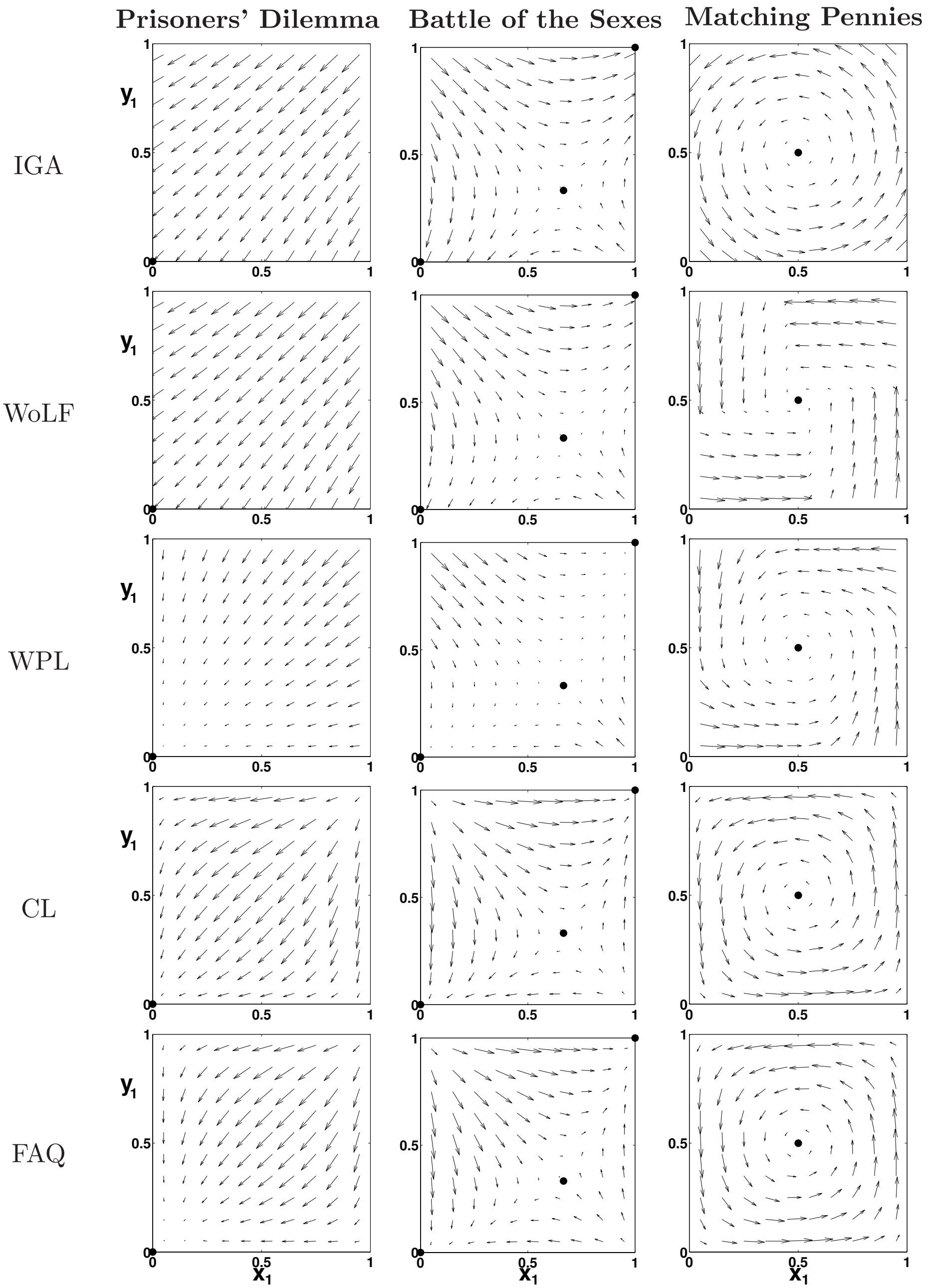


Figure 5.11: This figure shows the learning dynamics of the various algorithms in the Prisoners' Dilemma, Battle of the Sexes, and Matching Pennies. The dynamics of RM are visually indistinguishable from CL in this scenario. The Nash Equilibria are indicated with •. WoLF uses $\frac{\alpha_{min}}{\alpha_{max}} = 0.1$, and FAQ uses $\tau = 0.1$.

5.2.3 Generalization to normal form games

The previous subsection juxtaposes multi-agent reinforcement learning with gradient ascent in two-action games. This section presents the more general case of normal form games, where each player has a finite discrete set of actions, and $x = (x_1, x_2, \ldots, x_n)$ such that $\sum_i x_i = 1$ and $\forall x_i : 0 \leqslant x_i \leqslant 1$. The first constraint imposes $\sum_i \eth_i = 0$ on the gradient, where $\eth_i$ is the i^{th} component of the gradient, i.e., $\eth_i$ is the partial derivative of the value function with respect to x_i. Gradient ascent uses a projection function to ensure these constraints are fulfilled, while multi-agent reinforcement-learning algorithms inherently maintain that property, e.g., generating valid policies from value estimations that are iteratively improved.

The value function in n-player normal form games is defined as $V(x, \hat{y}) = \sum_i x_i f_i(\hat{y}) = x f(\hat{y})$, where $f_i(\hat{y})$ denotes the payoff of action i against the set of $n-1$ opponents following strategies $\hat{y} = (y^1, y^2, \ldots, y^{n-1})$. In two-player normal form games $f_i(y) = (Ay^T)_i$. The i^{th} element of the gradient can be calculated as the partial derivative of V with respect to x_i. Let e_i denote the i^{th} unit vector; the differential with respect to x_i can then be defined as δe_i. However, recall that IGA projects $x + \Delta x$ onto the tangent space of x. This update is equivalent to normalizing δe_i using the orthogonal projection function $\Phi(\zeta) = \zeta - \frac{1}{n} \sum_j \zeta_j$ [Sandholm, 2010]. The gradient that IGA uses according to Equation 2.8 in normal form games can be written as

$$\begin{aligned} \frac{\partial V(x, \hat{y})}{\partial x_i} &= \lim_{\delta \to 0} \frac{[x + \Phi(\delta e_i)]\, f(\hat{y}) - x f(\hat{y})}{\delta} \\ &= \Phi(e_i) f(\hat{y}) \\ &= f_i(\hat{y}) - \frac{1}{n} \sum_j f_j(\hat{y}). \end{aligned}$$

Using $u = (\frac{1}{n}, \ldots, \frac{1}{n})$, the expected update of IGA is $\dot{x}_i = \alpha\, [f_i(\hat{y}) - u f(\hat{y})]$. Note that this algorithm updates all actions simultaneously.

Let us now assume the gradient is not known, but needs to be sampled for one action at a time. A stochastic version of IGA with the same dynamics, but only one action being updated at a time, would necessarily sample all actions equally often. In other words, it would execute u while estimating the value of x, which means it is an off-policy algorithm. The expected update of stochastic IGA is $\dot{x}_i = \alpha u_i\, [f_i(\hat{y}) - u f(\hat{y})]$.

In a multi-agent learning setup, learning off policy would however make it impossible for the other player to learn about the first, because the first player is not executing the policy he considers best. Self-play illustrates the futility of learning off-policy against each other: $\dot{x}_i = u_i \left[e_i A u^T - u A u^T\right]$. Eventually, the player is going to use what has been learned off-policy, and switch from executing u to x, and any other player would be faced with a sudden change in his environment that may now be nothing like it was before. The other player may now completely disregard what he learned about playing against u, and restarts learning once the first player is on-policy. For symmetry reasons, this means also the first player can restart learning. In conclusion, multi-agent reinforcement learning needs to be on-policy, and therefore stochastic gradient ascent would need to sample on-policy. Sampling on-policy yields different update frequencies for each action resulting in the replicator dynamics $\dot{x}_i = x_i\,[f_i(\hat{y}) - xf(\hat{y})]$, which are at the core of multi-agent reinforcement-learning algorithms. It follows that multi-agent reinforcement learning implements on-policy stochastic gradient ascent, which in contrast to off-policy reasoning, is able to learn from discrete interactions in normal form games.

5.2.4 Discussion

The gradient ascent dynamics assumes that the gradient is known or can be computed by the agent. This assumption is typically not fulfilled in reinforcement-learning problems. The merits of gradient ascent dynamics are more theoretical—it makes convergence guarantees possible at the cost of stronger assumptions. Similar guarantees have also been derived for evolutionary models of independent multi-agent reinforcement learning. For example, the dynamics of FAQ-learning have been thoroughly analyzed in two-agent two-action games showing convergence near Nash equilibria (see Chapter 3). These guarantees either study newly derived variations as for FAQ-learning, or they draw on well established models from evolutionary biology, e.g., the cyclic behavior of the replicator dynamics is a well studied phenomenon [Hofbauer and Sigmund, 2002]. The findings presented in this section reveal the commonalities of gradient ascent and the replicator dynamics. REINFORCE algorithms [Williams, 1992] estimate the gradient of the reinforcement function from samples, and thus lies at the intersection of gradient ascent and the replicator dynamics. If these samples are taken off policy, single-agent behavior would follow the gradient ascent dynamics but agents would not exhibit their learned behavior, and would in multi-agent settings not learn anything meaningful about each other. Hence, any stochastic gradient ascent algorithm that should learn from discrete interactions of several agents needs to be on-policy, and will behave in essence equivalently to the replicator dynamics, e.g., the linear-reward-inaction variant of REINFORCE is equivalent to a simple learning automaton and thereby also equivalent to Cross learning [Williams, 1992], which in turn maps exactly to the replicator dynamics [Börgers and Sarin, 1997].

5.3 Summary

This chapter provides two new perspectives on multi-agent learning dynamics. First, an orthogonal visualization for time-dependent dynamical systems has been proposed, supporting efforts to study and design time-dependent parameters. An illustrative example of a two-agent two-action game was discussed, and the method has been shown to naturally reveal time-dependent properties of the system. Doing so facilitates designing parameters with a systematic approach rather than setting them in an ad hoc manner. Moving from the traditional time-collapsed view of trajectories to an evolving density perspective also opens up new ways of analyzing the behavior formally, e.g., a formal concept of density and *divergence* has been used to provide an auxiliary proof of convergence for FAQ-learning dynamics [Kianercy and Galstyan, 2012].

Second, this chapter relates two seemingly diverse families of algorithms within the field multi-agent learning: gradient ascent and independent reinforcement learning. The main contributions can be summarized as follows: First, the replicator dynamics are identified as the core building block of various types of independent reinforcement-learning algorithms, such as Cross learning, regret minimization, and Q-learning. Second, the learning dynamics of these algorithms are juxtaposed with variants of gradient ascent in two-player two-action games, highlighting the similar structure around the gradient of the expected reward. Third, multi-agent reinforcement learning is shown to implement on-policy stochastic gradient ascent in normal form games. Recognizing the connection to on-policy stochastic gradient ascent provides a basis for studying what is learnable with independent reinforcement-learning algorithms in multi-agent games. This framework may be taken as a basis for establishing lower bounds on performance in multi-agent games similar to Probably Approximately Correct Learning guarantees in single-agent learning. I leave this direction as an open problem for future research.

6 Applications

This chapter presents an evolutionary analysis of meta strategies in auctions and poker. I follow the methodology introduced by Walsh et al. [Walsh et al., 2002]: Meta strategies are evaluated in a competition of n agents, and the expected payoff to each strategy is captured in a heuristic payoff table as explained in Section 2.4.3. This table can subsequently be used to approximate the payoff in an infinite population that evolves according to the replicator dynamics. This methodology is most applicable in domains where players choose between a small number of known strategies. The evolutionary analysis reveals which combination of these strategies is most likely to prevail if the agents repeatedly interact, and in which constellations they dominate other strategies. This chapter is based on previously published work [Hennes et al., 2012; Ponsen et al., 2009].

6.1 The value of information in auctions

This section presents an analysis of the competitive advantage of price signal information for traders in simulated double auctions. Previous work has established that more information about the price development does not guarantee higher performance. In particular, traders with limited information perform below market average and are outperformed by random traders; only insiders beat the market. However, this result has only been shown in markets with a few traders and a uniform distribution over information levels. Here, additional simulations of several more realistic information distributions extend previous findings. Furthermore, the market dynamics are analyzed with an evolutionary model of competing information levels. Results show that the highest information level will dominate if information comes for free. If information is costly, less-informed traders may prevail reflecting a more realistic distribution over information levels.

Markets play a central role in today's society, and range from stock markets to consumer-to-consumer e-commerce [Angel, 2002; Bajari and Hortacsu, 2003]. Economic theory often starts from perfect competition as an idealized assumption about markets. It relies, among other characteristics, strongly on a symmetric information structure. All traders have access to the same information about price and quality of goods. Many, if not all, of today's markets do not meet this utopian assumption and thus valorize the access to information. Undoubtedly, information is an important factor that has influence on trading success or losses. Insiders are clearly able to use their information to outperform the market. However, the relation between information level and success is not trivial.

6.1.1 Related work

Market forecasters and fund managers are generally assumed to be well informed, though for the most part perform below market average. Cowles [Cowles, 1933] has been the first to study this phenomenon and reports that a group of trained forecasters performed 4% below market average during a period of 4.5 years. These findings have since been confirmed by multiple studies; for an overview, the interested reader may consult prior work [Kirchler, 2010; Tóth et al., 2007], in particular Malkiel [Malkiel, 2003], which reports on returns of actively managed funds over a period of 30 years—fewer than 15% of the funds outperformed the market.

Toth et al. [Tóth et al., 2007] study the relation between information and performance for traders with various information levels both in simulation and human experiments. Average-information traders perform below market level, while uninformed traders reach the market average; highly informed traders beat the market. These results suggest that if a trader has no inside information, trading based on current market prices (uninformed) is most sensible. Relying on outdated or average information has a negative impact on returns.

Prior work [Huber et al., 2008; Kirchler, 2010] has investigated whether this negative impact can be explained by behavioral patterns. In particular, the authors test the hypothesis that low performance of average-information traders is the result of overconfidence, i.e. overestimating the value of (possibly outdated) information. Results show that traders do not exhibit overconfidence and low returns are caused by the asymmetric information structure itself. Huber [Huber, 2007] offers the following explanation: during trends, foresight is clearly advantageous. When the trend reverses, the average-information trader trusting its information performs worst due to outdated information. Non-informed (random) traders are safe from these systematic mistakes and instead gain and lose in roughly equal measure.

The vast body of previous work [Huber, 2007; Huber et al., 2008; Kirchler, 2010; Tóth and Scalas, 2007; Tóth et al., 2006, 2007] has evaluated the advantage of information in markets from various perspectives. However, only markets with a limited number of agents and uniformly distributed information levels have been considered, and information was assumed to be free. This study investigates several more realistic information distributions in larger markets. Furthermore, the analysis is extended by studying an infinite population of traders using an evolutionary model and demonstrate the influence of the price of information on market dynamics.

Recent work on automated mechanism design employs a similar evolutionary analysis, but is based on a different auction-simulation platform that does not use dividends for driving the price signal [Phelps et al., 2006, 2010a,b]. A wider taxonomy of auctions is available [Friedman, 1993; Parsons et al., 2011].

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 or the New York Stock Exchange (NYSE) [Angel, 2002; McMillan, 1994]. In practice, there are 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, the focus is 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 the literature [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 principle forms are the clearing house or continuous operation. 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 some buyer is willing to pay more than a seller is asking for. This mode allows higher transaction rates at the cost of some allocative efficiency. Experiments in this section will use continuous operation mode, since it reflects the day-time operation mode of the NYSE [Angel, 2002].

Value of information

It is common sense that training and additional information should increase performance for any task. However, the value of information in markets is non-monotonic, i.e., having some information may be worse than having none.

To measure the value of information, experiments in auctions measure revenue. Since revenue is heavily dependent on market conditions dictated by the price signal, it is normalized to reflect the relative return. Assume trader i receives revenue r_i. The average profit $r_{avg} = \frac{1}{n}\sum_i r_i$ in a market is used to compute the relative market return $u_i = \frac{r_i}{r_{avg}} - 1$ for each trader.

Figure 6.1 shows relative market return over information levels in a market with $n = 10$ agents, one agent for each of 10 information levels, where level 0 represents

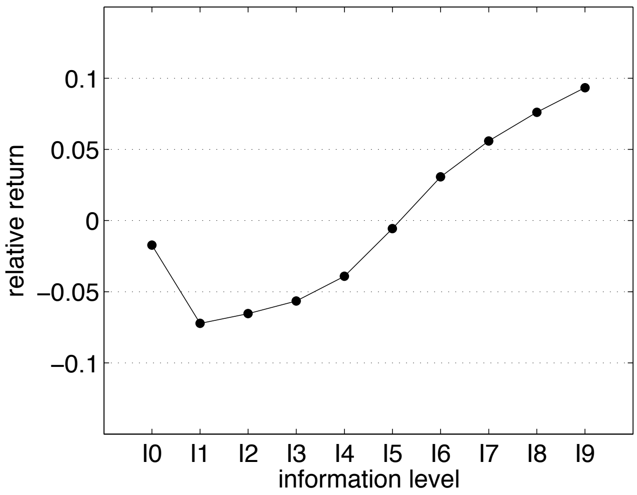


Figure 6.1: Relative market return over information level. 10 traders with information level 0 to 9 (1 trader for each level).

random traders (as formally defined in Section 6.1.2). The revenue follows a *J-curve*, which means that random traders perform at market average while weakly informed traders are exploited by insiders. This result holds both in abstract market models that can be simulated and in experiments with human participants [Tóth et al., 2007].

Previous research has demonstrated the viability of evolutionary game theory to analyze meta strategies in simulated auctions, and to compare clearing house against continuous double auctions [Kaisers et al., 2009; Phelps et al., 2005]. A similar analysis procedure is used here, but the data is generated by a different model described in the following section.

6.1.2 Market model

To analyze the advantage of foresight, a stock market is simulated with agents having different amounts of information on future prices, or *information levels*, trade a certain asset. This design closely follows the market model used in prior studies [Huber et al., 2008; Tóth and Scalas, 2007] to be comparable. The market is based on a continuous double auction with open order book, in which all traders can place bids and asks for shares. The intrinsic value of the shares is determined by a dividend stream that follows a random walk

$$D_t = D_{t-1} + \epsilon \tag{6.1}$$

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., $\epsilon \sim \mathcal{N}(0, 1)$. Figure 6.2 shows an example dividend stream.

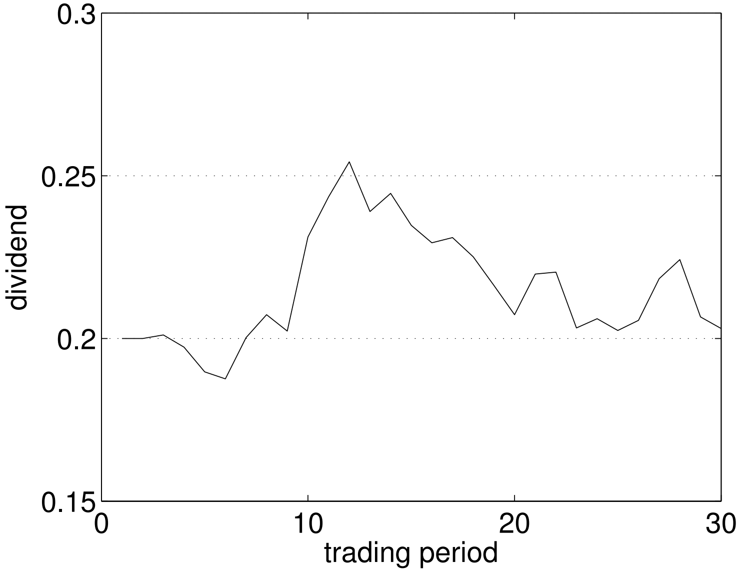


Figure 6.2: The dividend stream, following a Brownian motion.

The market is simulated over 30 trading periods, each lasting $10 \cdot n$ time steps, where n is the number of traders present. All traders start with 1600 units cash and 40 shares, each worth 40 in the beginning. At the beginning of each period, all traders can put a bid or ask in the book (opening call). Hereafter, at every time-step a trader is selected at random who can then place a bid or ask according to its trading strategy (see below). At the end of each period, a dividend is paid based on the shares owned, and 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., each share is valued according to the discounted future dividends (see below) and added to the cash reserves.

The different information levels are implemented by varying the amount of knowledge that traders have about the future dividends. In general, a trader with information level Ik knows the dividend of this and the next $(k-1)$ periods. Traders with information level I0 have no information about dividends and can only observe the current market price. This situation results in a cumulative information structure, where insiders know at least as much as average-information traders. The information that the traders receive each period is the conditional present value of the shares, conditioned on their information level. This value can be calculated using the dividend discount model (Gordon growth model) as

$$E(V|I_j, k) = \frac{D_{k+j-1}}{(1+r_e)^{j-2} r_e} + \sum_{i=k}^{k+j-2} \frac{D_i}{(1+r_e)^{i-k}} \tag{6.2}$$

where V denotes the value, I_j is the information level, k the period, and r_e the risk-adjusted interest rate (set to 0.5% in all experiments).

6.1.3 Trading strategies

Two different trading strategies are used in the experiments. Traders that have at least some information about the dividend stream (I1 and higher) use the *fundamentalist* strategy, that takes this information into account. Traders without any information (I0) use the *random* strategy, in which their bids and asks are based purely on the current market price of the shares.

Fundamentalists

Fundamentalists completely rely on the information they receive. The fundamentalist strategy is explained in Algorithm 1 [Tóth and Scalas, 2007; Tóth et al., 2006]. In essence, they compare their estimated present value $E(V|I_j, k)$ 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.

Algorithm 1 Fundamentalist trading strategy

```
$pv \leftarrow E(V|I_j, k)$ {private value}
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
    $placeAsk(pv + 0.25 \cdot \Delta_{ask} \cdot \mathcal{N}(0,1))$
  else
    $placeBid(pv + 0.25 \cdot \Delta_{bid} \cdot \mathcal{N}(0,1))$
  end if
end if
```

Random traders

The random trading strategy only takes the current market price into account when deciding whether to accept or place an order. With equal probability the trader sells or buys shares. The random trading strategy is explained in Algorithm 2. This algorithm is used to be consistent with previous work [Tóth and Scalas, 2007; Tóth et al., 2006]; however, results do not change if the Fundamentalist strategy with the market price as the private value is used instead of the random traders.

Algorithm 2 Random trading strategy

```
$\mathsf{pv} \leftarrow$ *current market price* {private value}
**if** $\mathcal{U}(0,1) < 0.5$ **then**
    $\mathsf{ask} = \mathsf{pv} + 2 \cdot \mathcal{N}(0,1)$
    **if** $\mathsf{ask} < \mathsf{bestBid}$ **then**
        $\mathsf{acceptOrder}(\mathsf{bestBid})$
    **else**
        $\mathsf{placeAsk}(\mathsf{ask})$
    **end if**
**else**
    $\mathsf{bid} = \mathsf{pv} + 2 \cdot \mathcal{N}(0,1)$
    **if** $\mathsf{bid} > \mathsf{bestAsk}$ **then**
        $\mathsf{acceptOrder}(\mathsf{bestAsk})$
    **else**
        $\mathsf{placeBid}(\mathsf{bid})$
    **end if**
**end if**
```

6.1.4 Evaluation of selected information distributions

The market is simulated with varying numbers of agents for each information level to analyze the relative performance of agents with different amounts of foresight. To reduce the effect of randomness, 100 sessions of 100 simulations each are performed; the dividend stream is fixed for each session. Results are given as the relative performance with respect to the market average plotted against the information levels.

Figure 6.1 (see page 94) shows the results for a market of 10 agents in 10 information levels: one random trader, I0, and 9 fundamentalists, I1 . . . I9. As can be seen, performance does not necessarily increase with more information: the random trader performs at market average, whereas traders with limited amounts of information do significantly worse. Only highly informed traders are able to beat the market.

This result is in line with related work, where a similar shaped *J-curve* was reported [Kirchler, 2010; Tóth and Scalas, 2007]. This relation between information level and performance, where more information is not always better, has also been observed in market experiments involving human traders [Huber et al., 2008]. A possible explanation is that random traders are by definition not predictable, and therefore hard to exploit by insiders. On the other hand, experts can more easily predict and exploit traders with limited or average information levels.

Previous work has mainly focussed on small scale markets, with uniform and static distributions of traders over information levels [Huber, 2007; Huber et al., 2008; Kirchler, 2010; Tóth and Scalas, 2007; Tóth et al., 2006, 2007]. This focus on the small may overly simplify reality, which may in turn influence the reported findings. For example, a market will be more likely to contain only a small number of insiders, and a large group of average-information traders. Furthermore, having only one trader per information level rules out within-group trading, which could bias results.

Previous studies [Kirchler, 2010; Tóth and Scalas, 2007] are extended here by looking at markets with more traders and non-uniform distributions of traders over information levels. An overview of relative market returns for a selection of information distributions is given in Figure 6.3. Next to the uniform distribution used in previous work, simulations with a normal distribution and a power-law distribution over information levels are evaluated. These distributions are chosen to reflect information distributions

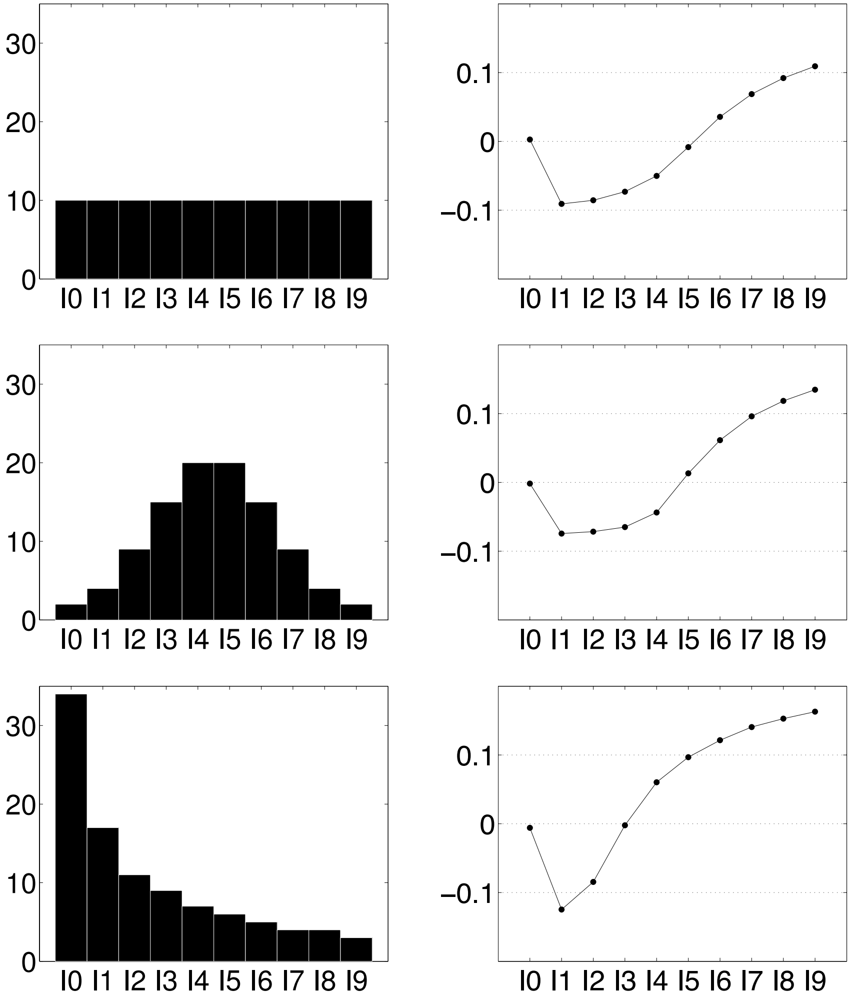


Figure 6.3: Relative market return over information level (right) for various information distributions (left) given a finite population of 100 traders.

that are likely to be found in real markets. It is impossible to observe these distributions directly as the information level is private to the trader [Huber et al., 2008]. The two chosen distributions follow from natural assumption: (1) normal distributions arise if access to information is cumulative based on independent and identically distributed bits of information; (2) the power-law distribution is motivated by a information flow in scale-free social networks where every trader has access to information of his social ties.

As can be seen from this figure, random traders perform at market average under all three distributions, and traders with limited information underperform the market. However, the shape of the curve does change considerably depending on the information distribution. Where in the uniform scenario only traders with information level I6 or higher outperform the market, for the normal distribution it is the case for I5 and for the power-law distribution for I4. However, the J-curvature is found to be relatively insensitive to changing information distributions and numbers of traders. Only in extreme cases (not shown here) does the curve change drastically.

Figure 6.4 shows that relative market returns follow the J-curve even in small markets with only three information levels: random traders, average-information traders, and insiders. Again, this finding is in line with previous work, where a similar setup was shown to reflect stylized facts such as autocorrelation observed within real markets [Tóth et al., 2007]. Note that the obtained curve does not change qualitatively when varying the information level of average-informed traders: any choice between $\{I0, I1, I9\}$ and $\{I0, I8, I9\}$ results in a J-curve.

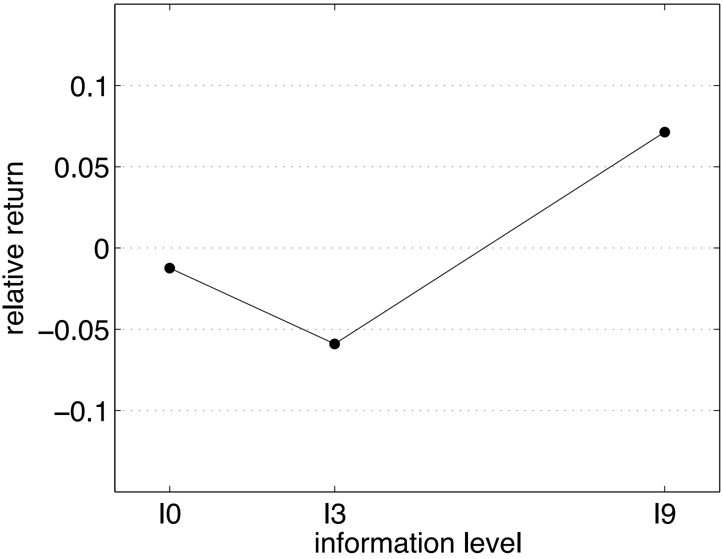


Figure 6.4: Relative market return over information level. 1 trader for each of the information levels 0, 3 and 9.

Discussion

The J-curve of relative market returns over information levels that has been observed in previous work has been reproduced. Furthermore, the specific shape of this curve also prevails if the distribution over information levels changes. This finding indicates that the conclusions drawn from this may hold under more realistic settings as well.

This perspective still assumes that the distribution over information levels does not change over time. There are several ways this assumption may be violated in practice: First, traders may choose to acquire more information or not. For example, traders may or may not subscribe to financial news sources, which in turn determines their information level—possibly at a cost. The effect of having traders choose between trading strategies has been investigated [Kirchler, 2010; Tóth and Scalas, 2007], with the conclusion that only highly informed traders will choose their fundamental strategy, taking their information into account. Second, traders may take over a larger market share due to their financial success while others are driven out of the market. This observation motivates the evolutionary analysis that accommodates evolving distributions in information levels, and elicits the market dynamics.

6.1.5 Evolutionary analysis

The previous section provides a method for computing expected relative market revenues for selected information distributions. This method views information distributions as isolated and fixed in time. However, the market revenue can be interpreted as Darwinian fitness, such that traders performing below market average should be driven out of the market, while those with higher returns prevail. This section will first introduce the evolutionary analysis methodologically, list the results and discuss their implications.

The evolutionary model assumes an infinite population. The payoff for such a population cannot be computed directly, but it can be approximated from evaluations of a finite population. For this purpose, the heuristic payoff table is used as described in Section 2.4.3. The expected payoff in an infinite population model can be computed from the heuristic payoff table and is used in Equation 2.6 to compute the evolutionary change according to the replicator dynamics.

Experimental setup and results

The experiments of this section comprise two elements. An evolutionary analysis of an infinite population is performed to elicit the dependence of revenue on the presence of other information levels. In addition, selected population distributions are approximated with a finite population and illustrate revenue distributions for interesting points. The evolutionary analysis is based on the market model described in Section 6.1.2 and uses the method described in the previous section to compute payoffs and the replicator dynamics for an infinite population of traders with arbitrary and evolving information distributions. The heuristic payoff tables are computed for $n = 12$ traders distributed over the information levels I0, I3 and I9, leading to 91 rows.

Figure 6.5 shows the evolutionary dynamics of the market model. Four representative population distributions are evaluated in more detail in a finite population of $n = 100$ traders to illustrate the revenue structure for the individual information levels. The evolutionarily stable state is a global attractor, where only insiders prevail. The relative market performance for the four selected finite distributions of traders highlights again the J-curve observed before. Even though uninformed traders perform close to market average, insiders take advantage of their knowledge and take over the market. However, their competitive advantage is vanishing as they are facing more and more competitors of the same information level (see top-right revenue graph of Figure 6.5).

Note that up till now, information was freely available to all traders. However, it is reasonable to assume that gathering more information is costly. In the most simple case, a fixed cost for information might be incurred leading to a possible advantage of uninformed traders as they do not have to pay this price. More realistically, costs could also increase with the amount of information gathered, for example using a quadratic cost function such that average-information traders pay only a little whereas insiders pay the full price. This scheme relates to a real-world scenario where average traders only subscribe to financial newspapers or magazines, whereas insiders may need to hire experts to gain their information advantage.

Figure 6.6 shows the market dynamics in both cost scenarios. The fixed cost is set to 5 units cash per trading period for information levels I3 and I9, uninformed traders

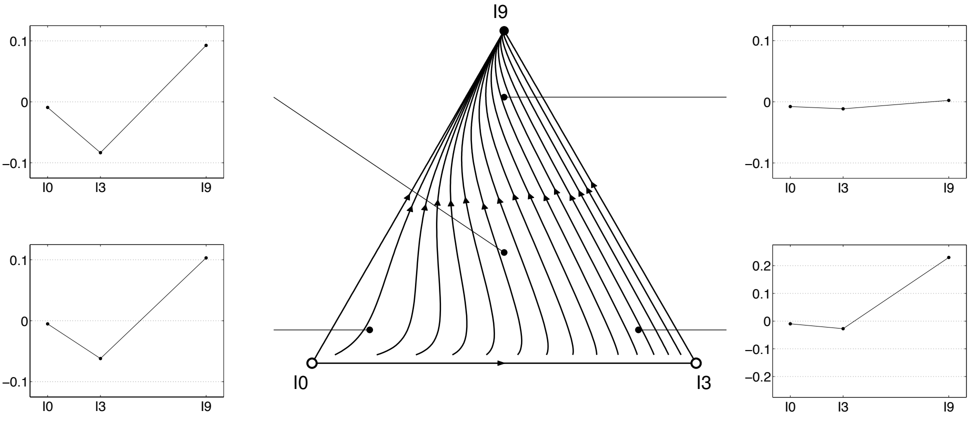


Figure 6.5: The central simplex shows the evolutionary dynamics of an infinite population mixing between the information levels 0, 3 and 9. Relative market revenue over information levels is given for four selected finite distributions: top-left (33, 33, 33) which reflects a uniform distribution, bottom-left (80, 10, 10), top-right (10, 10, 80), bottom-right (10, 80, 10).

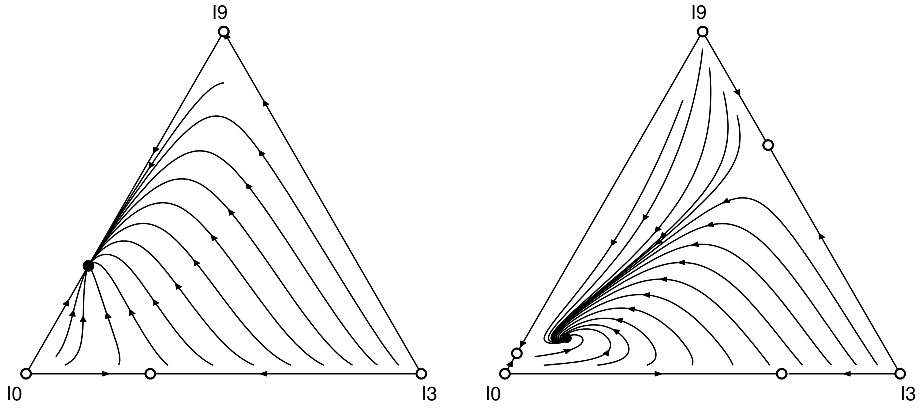


Figure 6.6: Evolutionary market dynamics using a fixed cost (left) and quadratic cost function (right) for information levels 0, 3 and 9.

pay nothing. The quadratic cost function used is

$$\frac{i^2}{9^2} \cdot 15,$$

where i is the information level, resulting in a maximum cost of 15 units cash for insiders per trading period. As can be observed, introducing cost leads to significantly different and more complex dynamics. In the constant cost scenario, the evolutionary advantage of insiders decreases in favor of uninformed traders, leading to an equilibrium state where insiders and uninformed traders co-exist. Using a quadratic cost function gives rise to an interior equilibrium, where all information levels prevail.

6.1.6 Discussion

Information does come at a cost in real markets, which has been neglected in much of the related work [Huber, 2007; Huber et al., 2008; Kirchler, 2010; Tóth and Scalas, 2007; Tóth et al., 2006, 2007]. Evolutionary analysis under different cost functions indicates that costs can significantly alter the market dynamics and allow less-informed traders to prevail.

The results contribute to the ongoing debate about the strong-form efficient-market hypothesis, which has a large following and growing number of critics [Fox, 2009]. It states that prices in financial markets instantly reflect all information available to participating traders, including insider information. Evolutionary pressure drives a market toward an information distribution at which the market is strong-form efficient, possibly driving some information levels extinct in the process. However, the evolutionary process will only end in equilibrium for an isolated system; in real markets, traders that enter the market with information and money from other sources continuously perturb the system. As a result, real markets may be found off-equilibrium almost all the time. It is up to future experiments to quantify the influence of arriving traders on perturbation from the equilibrium.

The literature has established a link between human traders and a market model that can be rigorously analyzed in simulation. In this section, this link has been exploited in the following ways: (1) The value of information in markets has been confirmed to follow a J-curve for several more realistic information distributions. (2) The evolutionary advantage of information makes insiders drive less-informed traders out of the market, with a diminishing competitive edge. (3) If information comes at a cost, less-informed traders may prevail in the market.

The experiments that have been carried out for this section were limited to an evolutionary analysis of three competing information levels. While this design choice is sufficient for demonstrating the arguments within this section, the evolutionary analysis naturally extends to four or more strategies. As such these findings pave the way for larger scale comparisons. In addition, future work may test the hypothesis that a market's informational efficiency is perturbed by traders moving in or out of a market.

6.2 Meta strategies in poker

In this section, the strategic interaction in the game of poker is analyzed by applying the evolutionary analysis to data gathered from a large number of real world poker games. This study uses two Replicator Dynamics models: First, the basic selection model is used to study the empirical data; second, an extended model that includes both selection and mutation is evaluated. These two models elicit the dynamic properties by describing how rational players and players with limited rationality switch between different strategies under different circumstances, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. The dynamics are illustrated using a simplex analysis. Experimental results confirm existing domain knowledge of the game, namely that certain strategies are clearly inferior while others can be successful given certain game conditions.

6.2.1 Introduction

Although the rules of the game of poker are simple, it is a challenging game to master. There are many books written by domain experts on how to play the game [Brunson, 1979; Harrington, 2004; Sklansky, 1987]. A general advice given to human players is that a winning poker strategy should be adaptive: a player should change the style of play to prevent becoming too predictable, but moreover, the player should adapt the game strategy based on their opponents. In the latter case, players may want to vary their actions during a specific game [Davidson et al., 2000; Ponsen et al., 2008; Southey et al., 2005], but they can also consider changing their strategy over a series of games (e.g., play a more aggressive or defensive style of poker).

In this section, an evolutionary game theoretic analysis of poker strategies is applied to data from real world poker games played between human players. More precisely, I investigate the strengths of a number of poker strategies facing some opponent strategies using Replicator Dynamics (RD) models [Hofbauer and Sigmund, 2002; Maynard Smith, 1982; Taylor and Jonker, 1978; Tuyls et al., 2006]. Replicator dynamics are a system of differential equations describing how strategies evolve through time. Here, two of such models are examined. The first RD model only includes the biological selection mechanism. Studies from game theory and reinforcement learning indicate that people do not behave in purely greedy and rational ways in all circumstances but also explore different available strategies (to discover optimal strategies) for which they are willing to sacrifice reward in the short term [Gintis, 2009; Sutton and Barto, 1998]. It is thus critical to include mutation as an exploration factor to the RD model to find accurate results. To account for exploration, a second RD model that includes both selection and mutation terms is applied.

Several heuristic strategies are defined, i.e., strategic behavior over large series of games, and a heuristic payoff table is computed that assigns payoffs to each of these strategies. This approach has been used to analyze the behavior of buyers and sellers in automated auctions, e.g., as presented in Section 6.1 or in the literature [Phelps et al., 2004; Vytelingum et al., 2007; Walsh et al., 2002], and it is described in detail in Section 2.4.3. Conveniently, for the game of poker, several heuristic strategies are already defined in the poker literature and can be used in the analysis.

The innovative aspects of this study are twofold: First, although there are good classical game-theoretic studies of poker, they are mainly interested in the static properties of the game, i.e. what the Nash equilibria are and how to explicitly compute or approximate them. Due to the complexity of this computation, usually simplified versions of poker are considered [Billings et al., 2003]. Instead, here an evolutionary perspective sheds light on this game using two different RD models. This use of RD enables the investigation of the dynamic and interactive properties of play by studying how rational players switch between different strategies when faced with a certain composition of competing strategies. In addition, study of the dynamics reveals the basins of attraction of the equilibria, and what the stability properties of the attractors are. These new insights help to unravel the complex game of poker and may prove useful for strategy selection by human players, but can also aid in creating strong artificial poker players. Second, this analysis is based on real world data that is obtained by observing poker games at an online website, wherein human players competed for real money at various stakes. From this real world data, the heuristic payoff table is derived, as opposed to the artificial data used in the previously mentioned auction studies. By analyzing real world data, the claims put forward by domain experts on the issue of strategy selection in poker can be validated empirically.

The remainder of this section is structured as follows. First, the specific poker variant under examination is explained, namely No-Limit Texas Hold'em poker, and some well-known strategies for this game are described. Next, I elaborate on the RD method and continue with a description of the methodology. Finally, experiments are presented and discussed, and the section closes with some conclusions.

6.2.2 Background

In the following, the rules of the game of poker are briefly summarized. Subsequently, I list several ways of categorizing poker strategies according to domain experts.

Poker

Poker is a card game played between at least two players. In a nutshell, the objective in poker is to win games (and consequently win money) by either having the best card combination at the end of the game, or by being the only remaining active player. The game includes several betting rounds wherein players are allowed to invest money. Players can remain active by at least matching the largest investment made by any of the other players, or they can choose to fold (i.e., stop investing money and forfeit the game). The winner receives the money invested by all the players within this game. In practice, players can join and leave tables, and at each table games are played with a group of players that remains the same for many games. Many variations have been devised concerning both the rules of the game as well as the rules for joining and leaving a table.

This study analyzes the strategic elements within the most popular poker variant, namely No-Limit Texas Hold'em. This game includes 4 betting rounds (or phases), respectively called the pre-flop, flop, turn and river phases. During the first betting round, all players are dealt two private cards (usually refered to as a player's *hand*) that are only known to that specific player. To encourage betting, two players are obliged to invest a small amount the first round (the so-called small- and big-blind). One by one, the players can decide whether or not they want to participate in this game. If they indeed want to participate, they have to invest at least the current bet. This investment is known as *calling*. Players may also decide to *raise* the bet. If they do not wish to participate, players *fold*, resulting in loss of money they bet thus far. A betting round ends when no outstanding bets remain, and all active players have acted. During the remaining three betting phases, the same procedure is followed. In every phase, community cards appear on the table (respectively, 3 in the flop phase, and 1 in the other phases). These cards apply to all the players and are used to determine the card combinations (e.g., a pair or three-of-a-kind may be formed from the player's private cards and the community cards). After the last betting round, the card combinations for active players are compared during the so-called showdown.

Classifying poker strategies

There is a vast body of literature on winning poker strategies, mostly written by domain experts [Brunson, 1979; Harrington, 2004; Sklansky, 1987]. These poker strategies may describe how to best react in detailed situations in a poker game, but also how to behave over large numbers of games. Typically, experts describe poker strategies (i.e., behavior over a series of games) based on only a few aggregate features. For example, an important feature in describing a player's strategy is the percentage of times this player voluntarily invests money during the pre-flop phase and then sees the flop (henceforth abbreviated as *VPIP*), since this may give insight into the player's hand. If a particular player sees the flop more than, let's say, 40% of the games, he or she is likely playing with low quality hands [Sklansky, 1987] compared to players that only see the flop rarely. The standard terminology used for respectively the first approach is a *loose* and for the latter a *tight* strategy.

Another important feature is the so-called *aggression-factor* of a player (henceforth abbreviated as *AGR*). The aggression-factor measures whether a player plays offensively (i.e., bets and raises often), or defensively (i.e., calls often). This aggression factor is calculated as a ratio between the fractions of a player's decision to bet, raise or call:

$$\frac{\%\text{bet} + \%\text{raise}}{\%\text{calls}}$$

A player with a low aggression-factor is called *passive*, while a player with a high aggression-factor is simply called *aggressive*.

6.2.3 Methodology

This section outlines the methodology of the analysis, and refers to other sections for the formal description of replicator dynamics and the heuristic payoff table. I recap how the replicator dynamics are combined with the heuristic payoff table that is used to derive average payoffs for the various poker strategies.

The replicator dynamics [Taylor and Jonker, 1978; Zeeman, 1981] are a system of differential equations describing how strategies evolve through time. It assumes an infinitely large population of "individuals" (i.e., players). Each player may apply one of the available "replicators" (i.e., strategies). The pure strategy i is played with probability x_i, according to the vector $x = (x_1, \dots, x_k)$. The profit of each player depends on the population composition x. The payoff to each heuristic poker strategy in a composition of a finite population is captured in a heuristic payoff table and used to estimate the payoff in the infinite population model as described in Section 2.4.3. At each time step, players may switch their strategies based on the profits received (i.e., they switch to more successful strategies). As a consequence, the probabilities of strategies are changed. This adaptation is modeled by the replicator dynamics from evolutionary game theory.

An abstraction of an evolutionary process usually combines two basic elements: selection and mutation. Selection favors some population strategies over others, while mutation provides variety in the population. In this research, two replicator dynamics models are considered. The first one is based solely on selection of the most fit strategies in a population. The second model, which is based on Q-learning [Tuyls et al., 2006, 2003], includes mutation in addition to selection terms. The RD are given in Section 2.4.2, and the RD with mutation are described by the idealized model of Q-learning, introduced in Section 2.5.3. For all described selection-mutation experiments, the mutation parameter τ is fixed at 0.1.

6.2.4 Experiments and results

The evolutionary analysis is based on a collection of 318535 No-Limit Texas Hold'em games played by a total of 20441 human players at an online poker site. The data features tables with varying numbers of players participating in a single game, ranging from two-player games to full-table games with 9 players. As a first step, the strategy for each player in any given game needs to be classified. If a player played fewer than 100 games in total, the data is considered insufficient to establish a strategy classification, and the player and respective games are ignored. If the player played at least 100 games, intervals of 100 games are used to collect statistics for this specific player; these statistics then determine the *VPIP* and *AGR* values (see Section 6.2.2). The player's strategy is then labelled according to these two values, and the resulting strategy classification is associated with the specific player for all games in the interval. Having estimated all players' strategies, it is now possible to determine the discrete profile (i.e., the number of players playing any of the available strategies) for all games. This discrete profile assigns each game to a row in the heuristic payoff table. Finally, the average payoffs for all strategies given a particular discrete profile can be computed.

Next, I highlight several experiments with varying strategy classifications. Of course, more complex strategy classifications are possible, but the ones chosen are often used by domain experts and serve as a good starting point to keep the analysis of their interplay tractable.

Analyzing pre-flop and post-flop play

The first two experiments examine pre-flop and post-flop play in isolation. To be more specific, each player's strategy is labeled solely based on either their *VPIP* or *AGR* values. Table 6.1 gives the rules for the strategy classification. These rules were derived from domain knowledge and are common for classifying strategies in a No-Limit Texas Hold'em game [Brunson, 1979; Harrington, 2004; Sklansky, 1987].

The *VPIP* determines the pre-flop strategy, and gives insight in the player's card selection. A loose player plays a wider range of cards whereas a tight player will wait for more quality cards (i.e., those that have a higher probability of winning the game at showdown when cards are compared). The *AGR* value determines the post-flop strategy, and denotes the ratio between aggressive (i.e., betting and raising) and passive (i.e., calling) actions.

It is often claimed by domain experts that aggressive strategies dominate their passive counterparts. The rules of the poker game, and in particular the fact that

Table 6.1: Strategy classification for pre-flop and post-flop play in poker.

pre-flop	Rule	post-flop	Rule
Tight	$VPIP < 0.25$	Passive	$AGR < 1$
Semi-Loose	$0.25 \leqslant VPIP < 0.35$	Neutral	$1 \leqslant AGR < 2$
Loose	$VPIP \geqslant 0.35$	Aggressive	$AGR \geqslant 2$

games can be won by aggressive actions even when holding inferior cards, seem to back up this claim. Figure 6.7 examines the strategic advantage of strategies with varying VPIP values. Figure 6.7a (selection) yields one strong attractor that lies at the pure strategy AGGRESSIVE. Figure 6.7b (selection-mutation) shows a mixed equilibrium strategy mainly between AGGRESSIVE and NEUTRAL. Again, the AGGRESSIVE strategy is played 3 out of 4 games. These results confirm the claim that aggressive strategies generally dominate passive ones.

For the pre-flop strategy, the tight strategy is often assumed to be best, in particular for less skillful players. However, it is also claimed that the pre-flop strategy should depend on the strategies played by the opponents. If the majority of players play a tight strategy, then a looser strategy pays off and vice versa.

Figure 6.8a (selection) features an attractor lying in the pure strategy TIGHT. Similarly, he selection-mutation model in 6.8b yields a mixed strategy between TIGHT and SEMI-LOOSE. Still, the TIGHT strategy is dominant and is played 8 out of 10 games. These findings seem to contradict the claim that one should mix their pre-flop play according to the opponent strategies. However, this perspective does not explicitly differentiate based on the post-flop strategies. The previous experiment has already shown that aggression is a key strategic choice for the utility of the overall strategy. A more differentiated evaluation of Figure 6.8 suggests that the TIGHT strategy is optimal in expectation, i.e., given a random post-flop strategy from the observed distribution. Mixed strategies in pre-flop play may become rational when players use very specific post-flop strategies, e.g., always playing aggressive after the flop.

Analyzing Complete Poker Strategies

The next series of experiments combines both *VPIP* and *AGR* features for strategy classification. The rules used are shown in Table 6.2. Again note that these strategy classifications are derived from the poker literature, although here the number of at-

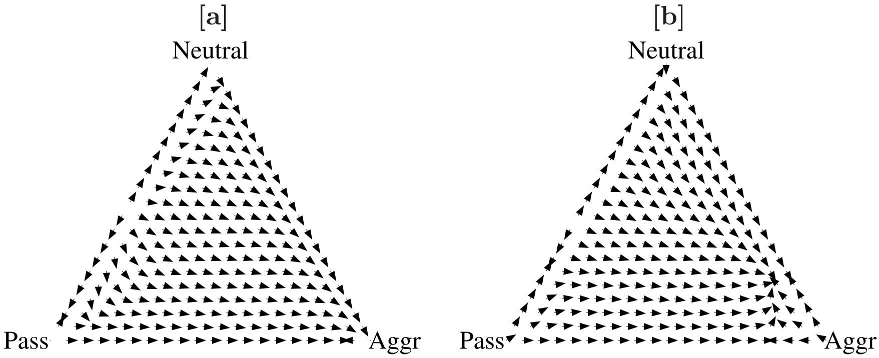


Figure 6.7: Dynamics of post-flop strategies using the replicator dynamics based on selection (a) and selection combined with mutation (b)

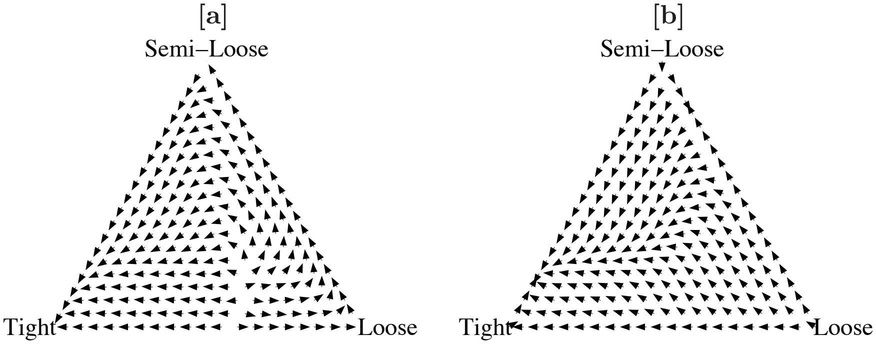


Figure 6.8: Dynamics of pre-flop strategies using the replicator dynamics based on selection (a) and selection combined with mutation (b)

tributes per feature is reduced to two, resulting in exactly four strategies: tight-passive (a.k.a. Rock), tight-aggressive (a.k.a. Shark), loose-passive (a.k.a. Fish) and loose-aggressive (a.k.a. Gambler).

Experts argue that the Shark strategy is the most profitable strategy, since it combines patience (waiting for quality cards) with aggression after the flop, while the Fish strategy is considered the worst possible strategy.

Recall from Section 2.4.2 that each simplex shows the competitive strategic advantage of three strategies. For this experiment, a total of four strategies is available to the players. Hence, one strategy is excluded per plot by only considering discrete profiles from the heuristic payoff table where no players chose the excluded strategy. This results in four different combinations of three strategies. Here, trajectories are used to illustrate the dynamics for both the selection and selection-mutation model.

Figure 6.9a, Figure 6.10a and Figure 6.12a confirm that both passive strategies, i.e., the Fish and Rock strategies, are dominated by the two aggressive strategies Shark and Gambler. Furthermore, the attractors in Figure 6.9a and Figure 6.10a lie close to the Shark strategy; this strategy is played with 80% and 65% probability respectively. In Figure 6.12a, the Gambler strategy is slightly preferred over the Shark strategy, which is played 40% of the time. These results imply that Shark is a strong strategy, as was suggested by domain experts. Only in Figure 6.12 is Shark slightly dominated by Gambler. Similarly, the Fish strategy is a repeller, with the exception of Figure 6.11, where the equilibrium is mixing Fish with the Rock strategy.

Table 6.2: List of meta-strategies and rules for strategy classification in poker.

Strategy	Rule
Rock	$VPIP < 0.25$, Passive $AGR < 2$
Shark	$VPIP < 0.25$, Passive $AGR >= 2$
Fish	$VPIP >= 0.25$, Passive $AGR < 2$
Gambler	$VPIP >= 0.25$, Passive $AGR >= 2$

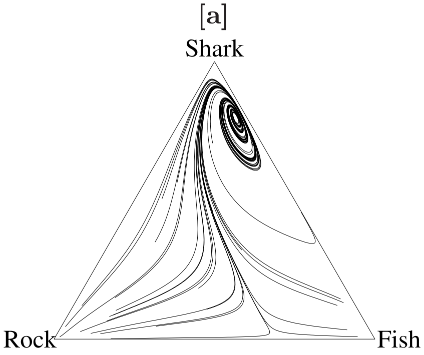


Figure 6.9: Trajectory plots analyzing the Rock, Shark and Fish strategies using the RD based on selection (a) and selection-mutation (b)

The selection-mutation plots show similar results with mixed strategies close to the Shark. In general, the equilibria found through selection-mutation lie closer to the center of the simplex and therefore mix more between the available strategies. This behavior is inherent to the selection-mutation model, which includes players' exploration of all available actions. An interesting observation in Figure 6.9 is that for the mixed strategy using the selection model the Fish strategy is played more compared to the Rock strategy (respectively 17% to 3%), while the selection-mutation model suggests the opposite. Here, the Rock strategy is played more with 17% to 10%. Domain experts believe the Fish strategy is inferior over all other strategies. Thus, results from the selection-mutation model align better with expert advice.

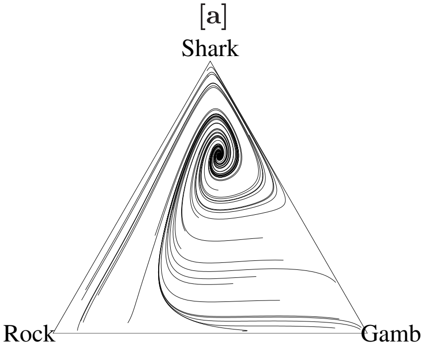


Figure 6.10: Trajectory plots analyzing the Rock, Shark and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

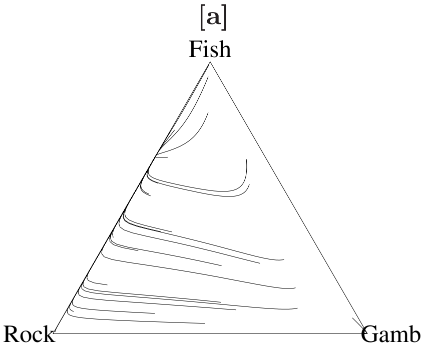


Figure 6.11: Trajectory plots analyzing the Rock, Fish and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

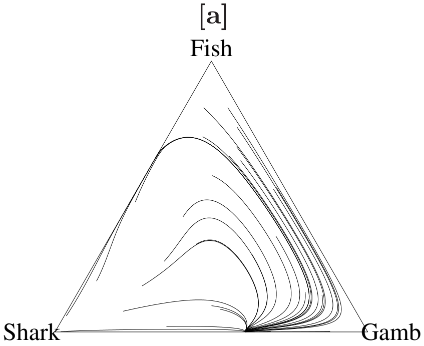


Figure 6.12: Trajectory plots analyzing the Shark, Fish and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

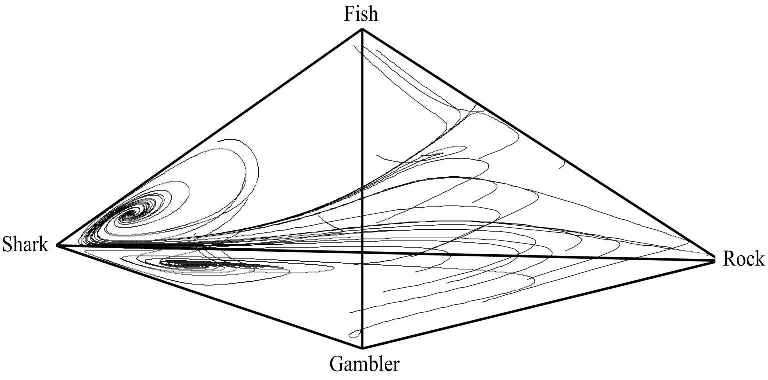


Figure 6.13: Trajectory plots in 3-dimensional space analyzing dynamics for all 4 strategies

A shortcoming of the leave-one-out approach is that it only captures the boundary conditions of the true strategy space mixing between four strategies. Therefore, the following experiment analyzes the dynamics among all four strategies at once. The result for the selection model is represented in Figure 6.13 as a 2-dimensional representation of the 3-dimensional space. Several random interior points were chosen to visualize the basins of attraction by their trajectories. The dynamics are similar to the previous plots, but there are differences. For example, only two attractors remain both near the Shark strategy, an the attractor found in Figure 6.12a is not attracting interior points. The attractors near the Shark strategy clearly have a stronger basin of attraction, i.e. trajectories are more likely to end up in one of these equilibria.

For the experiments with selection-mutation dynamics, the results are summarized numerically. It yields only one attractor near the mixed strategy $56\%, 25\%, 17\%$ and 2%, for respectively the Shark, Rock, Gambler and Fish strategy. The Fish strategy effectively goes extinct under this model.

6.2.5 Discussion

This case study has investigated the evolutionary dynamics of strategic behavior in the game of No-Limit Texas Hold'em poker. The evolutionary game theoretic perspective reveals how rational players switch between different strategies under different competition, in particular using two Replicator Dynamic models, one that is purely driven by selection, and another that also contains mutation. The analysis is based on observed poker games played at an online poker site, and identifies several heuristic poker strategies based on domain knowledge. The payoff to each strategy under various opponent strategies is captured in the heuristic payoff table, and further used in the context of the replicator dynamics. The results have been visualized in simplex plots that show where the equilibria lie, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. The results mainly confirm expert advice, namely that aggressive strategies mostly dominate their passive counterparts. Overall, the selection-mutation model reflected what domain experts claim even more closely than the basic model of selection.

Future work shall examine the interactions between the strategies among several other dimensions. For example, one could look at more detailed strategy classifications (i.e., based on more features) or represent strategies in a continuous way.

6.3 Summary

This chapter has demonstrated the evolutionary game theoretic approach on two applications. First, the value of insider information in double auctions has been studied. The results have shown that a little information may be significantly worse than having no information at all, but overall insiders dominate the market if information is free. This result of the evolutionary analysis confirms previous findings that the value of information follows a J-curve. Additional experiments have revealed that if information comes at a cost, less informed traders may prevail. The more complex population model in combination with costs of information extends the state-of-the-art methodology to analyze the value of information in double auctions. In future work, this method can be extended further to study auctions with exogenous events to capture more realistic scenarios, such as trader in- and outflow.

Second, the analysis has been applied in the domain of poker. The rigorous analysis has confirmed conventional poker wisdom, namely that aggressive strategies dominate their passive counterpart in most settings. In addition, the explorative models including mutation match even better with human expert advice, suggesting that they are a better model for human behavior in poker.

7 Conclusions

The previous chapters have discussed each of the contributions in detail. This chapter concludes the dissertation with a discussion of the contributions and how these answer the research questions that were set forward in the introduction. In addition, some limitations of the presented approach are discussed and promising avenues for future research are pointed out.

7.1 Answers to the research questions

Seven research questions were put forward in Section 1.4 and will each be answered explicitly based on the findings presented in the corresponding chapters.

1. Why does Q-learning deviate from the idealized model, and how can Q-learning be adjusted to show the preferable behavior of the idealized model? *Chapter 3*

   Section 3.1 has described why Q-learning deviates from the idealized model. The Q-learning algorithm updates one estimated action-value at a time, namely the one corresponding to the last selected action. As a result, the expected update of an action-value also depends on the frequency of the updates, and not only on the expected change given an update has occurred. Since only the estimate of the selected action i is updated, the frequency is determined by the probability x_i of the agent playing that action. The idealized model on the other hand has been derived under the simplifying assumption of updating all action-values at each time step. Thus, the expected behavior of Q-learning is equivalent to x_i times the dynamics of the idealized model. In experiments, this discrepancy results in policy updates that depend on the initialization of Q-values as pessimistic, neutral or optimistic, while the idealized model has no such dependencies.

Using these insights, the variation *Frequency Adjusted Q-learning* (FAQ-learning) has been introduced in Section 3.2 and its expected behavior closely adheres to the idealized model. This algorithm scales each update inversely proportionally to the update frequency of the corresponding action, thereby approximating the effect of an equal number of updates for all actions. Experiments have shown that FAQ-learning is consistent across Q-value initializations and indeed inherits the behavior of the idealized model.

2. What is the long term behavior of this idealized Q-learning model; does Q-learning converge to Nash equilibria? *Chapter 3*

   The idealized model that describes the dynamics of Frequency Adjusted Q-learning has been proven to converge to stable points in two-agent two-action games in Section 3.3. These stable points can be moved arbitrarily close to Nash equilibria by selecting an appropriately small exploration parameter. For high exploration, the stable point can be moved arbitrarily close to the uniform policy $(\frac{1}{2}, \frac{1}{2})$, and approaches the Nash equilibrium as exploration is decreased if there is only one. In case of Battle-of-Sexes type games, i.e., those featuring three Nash equilibria, the single stable point near the uniform policy shows a pitchfork bifurcation at a critical exploration rate and two attracting fixed points as well as one repelling fixed point appear for low exploration. The fixed points approaching pure equilibria are attracting, and the fixed point approaching the mixed equilibrium is repelling. This proof of convergence provides the first conclusive evidence of a variation of individual Q-learning to converge in multi-agent games. The auxiliary proof developed by other authors supports my findings, but does lack the grounding in the theory of FAQ-learning provided in Section 3.2 and 3.3.

3. How can the evolutionary framework be extended to more realistic scenarios such as varying exploration rates or multiple states? *Chapter 4*

   First, the learning dynamics of Frequency Adjusted Q-learning (FAQ-learning) have been extended to exploration rates that may vary with time. This extended dynamic model simplifies to the previously derived idealized dynamics if the exploration function is constant over time, i.e., its derivative is zero. This result also implies that given infinite time, the exploration may be decreased arbitrarily slowly, such that the derivative can be moved arbitrarily close to zero. As a consequence, the derived model is more relevant for modeling practical solutions of finite time than to provide convergence proofs, in which infinite time may be assumed. Second, the derivations of the evolutionary model have been extended to multi-state settings. These extended models show that the dynamics of both FAQ- and Q-learning depend on the Q-values in multi-state environments and cannot be reduced to a representation in the policy space. In other words, the interactive learning system is inherently high dimensional, which poses challenges for inspecting or analyzing such a complex system. Third, the dynamics of Lenient FAQ-learning have been examined and have shown that leniency makes it possible to increase the basin of attraction of the global optimum at the cost of speed of convergence.

4. Are there alternative perspectives on the time varying dynamics of multi-agent learning that enable a systematic design of time-dependent parameters? *Chapter 5*

   An orthogonal visualization of the learning trajectories has been proposed; it facilitates a systematic design of time-dependent parameters of both games and agents. This new perspective elicits the information of the learning system encoded in the density of joint policy particles, rather than only the development of one learning trajectory. Using a case study, this density-based method has been demonstrated by successfully setting a time-dependent parameter in an example application.

5. What are the commonalities and differences between variations of infinitesimal gradient ascent and the replicator dynamics? *Chapter 5*

   The dynamics of several variants of infinitesimal gradient ascent have been compared to the replicator dynamics in normal form games, and they share the gradient of the reward as a common building block in their dynamics. Infinitesimal gradient ascent assumes that information about the update of all actions is available at each time step, while the replicator dynamics relate to reinforcement-learning algorithms that only sample one action value at a time, and thus in expectation update proportionally to the frequency of selecting an action. Hence, reinforcement learning can also be considered stochastic gradient ascent on the payoff function, where updates are only applied to the sampled action.

6. How can the evolutionary analysis be applied in realistic domains, and more specifically what does it reveal about auctions and poker? *Chapter 6*

   The evolutionary analysis can be applied to practical domains by collecting payoffs in a heuristic payoff table, and computing the expected payoff for the evolutionary model therefrom. In this way, payoffs have been aggregated from simulations of auctions and from real world poker games. First, the value of information in auctions has been studied for a number of realistic information distributions between the traders. Results confirm previous findings, stating that the value of information is not monotonically increasing as information cumulates but rather follows a J-shape, i.e., having some information may be worse than having none, while only insiders outperform the market. Due to this advantage of insiders, these would drive lower-information traders extinct if information comes for free. If information is costly, lower-information traders may prevail. It should be noted that exogenous events have not been accounted for and may be another reason for the prevalence of lower-information traders.

> The analysis of poker strategies has mostly confirmed conventional expert advice, namely that aggressive strategies commonly dominate their passive counterparts. In addition, the evolutionary models including exploration have been found to be a better match to expert advice than those without exploration. This finding suggests that experts include aspects of exploration in their model of human behavior.

Overall, these six research questions have touched upon several key aspects of multi-agent learning: (1) the critical evaluation of state-of-the-art models and the firm establishment of the link between reinforcement-learning algorithms and dynamical systems, (2) the use of this link to prove convergence behavior, (3) the extension of the evolutionary framework to cover more complex learning algorithms and games, (4-5) improving the coherence of the evolutionary framework, and (6) leveraging insights from theory in applications.

7.2 Contributions to multi-agent learning agendas

The objective of single-agent learning is well-defined as optimal performance in the limit, fast speed of convergence and minimal or no regret [Kaelbling et al., 1996]. Multi-agent learning on the other hand is a younger field with more diverse ambitions, e.g., Shoham et al. have defined five agendas [Shoham et al., 2007]: (1) computing properties of a game, (2) describing natural agents, e.g., human learning, (3) determining whether algorithms are in equilibrium with each other (normative agenda), (4) prescribing distributed control that has desirable aggregate behavior, and (5) prescribing individually rational behavior. Empirical comparisons in benchmark problems provide a first guideline on the comparative performance of each algorithm [Busoniu et al., 2008; Panait and Luke, 2005]. However, these *black box* comparisons do not facilitate deep understanding of the strengths and weaknesses of learning algorithms, neither do they provide *guarantees*. This lack of a formal foundation makes it hard to generalize beyond the tested environments. In contrast, the link between dynamical systems and learning makes it possible to apply tools from dynamical systems to the analysis of multi-agent reinforcement learning, and allows a more rigorous study of interactions and parameter sensitivity. Chapter 3 contributes to the normative agenda by showing how the link to dynamical systems can be used to provide convergence guarantees by analyzing learning dynamics in the policy space. The analysis has been extended to more realistic scenarios in Chapter 4, throughout which results have been discussed in light of both prescriptive agendas (optimality with respect to both the system as well as the individual). Chapter 5 has provided a broader view at these dynamical systems, which are a capable tool for the pursuit of diverse goals—be they computational, normative, or prescriptive.

7.3 Limitations of the methodology and future work

Evolutionary dynamics have been used as a framework for multi-agent learning throughout the dissertation and yield many new insights, especially into the interactive influences of multi-agent learning. However, this approach as any other is tailored to a specific purpose with its own limitations in what it can deliver. More specifically, multi-agent reinforcement learning is a *stochastic* system that is subject to *time constraints* in any practical application. Taking the infinitesimal limit of the learning rate, it is linked to a *deterministic* dynamical system defined by partial differential equations. Since any single update is now infinitesimally small, any positive change relates to an infinite number of updates that take *infinite time* and these dynamical systems can thus only be seen as an abstract model of reality. Moving towards dynamical systems enables using tools like the eigenvalues of the Jacobian to determine the stability and convergence properties of the system and hence facilitates formal analysis. At the same time, it captures only the expectation of the stochastic system, and this approximation may create artifacts by itself, e.g., if the stochastic system is equally likely to go either way the expectation may suggest stability for specific boundary conditions. These limitations are inherent to the way in which the dynamical system is derived from the stochastic algorithms but need to be tolerated as limitations of this otherwise powerful approach.

Some notes on the limitations of the research method and presented experiments are pointed out below. There are many promising extensions that have not made it into this dissertation but could be explored in future work.

Chapter 3 The derivation of Frequency Adjusted Q-learning (FAQ-learning) closes a gap that becomes apparent when comparing the idealized model to Q-learning. The chapter concludes with a proof of convergence for FAQ-learning, which has subtle but essential differences from Q-learning by scaling each update anti-proportionally to the frequency of the action. If each update was scaled by the same factor, dynamical systems theory states that the qualitative behavior must be equivalent [Hofbauer and Sigmund, 2002], and the proof would transfer directly to Q-learning. However, since the factor depends on each action and player, this proof is only indicative rather than conclusive about the Q-learning convergence. A formal argument for the transfer to hold is still missing.

Chapter 4 The time-varying multi-state dynamics extend the dynamical system framework to more realistic settings. However, the interactive learning dynamics are inherently so complex that they become intractable to handle with current methods. More generally, proving convergence of high dimensional practical system (multi-state, many action, many agents) is challenging if not impossible. Here, empirical evaluation still yields more conclusive insights. In addition, the presented case study and design of a specific exploration function could be improved in several ways, e.g., the requirements of multi-agent learning could be linked to the formal requirements for exploration in single-agent learning. The model of leniency pushes the frontier of algorithms that are covered by the evolutionary framework. Additional experiments are required to validate whether the improved convergence to global optima justifies the increased time it takes to reach these optima.

Chapter 5 The new perspectives complement existing literature on how to capture these high dimensional learning dynamics. However, the orthogonal approach still needs to be grounded in established work, e.g., relating the initial distribution to a prior as it is used in Bayesian reasoning. In addition, both new perspectives would benefit from being demonstrated in applications. Overall, the theory of two-agent two-action games is quite ahead of the application of one-population models, and more advanced concepts, e.g., the new perspectives, need to be transferred and applied to practical problems.

Chapter 6 The underlying methodology of the applications chapter computes the evolutionary model based on heuristic payoff tables, assuming that individuals of an infinite population meet each other for competition in a finite small group. This model matches very well with the poker domain, in which a table is played and money is won or lost in competition to a small number of opponents. For other applications like auctions this model may be more of an approximation, since real systems may yield far more agents than evaluated in the heuristic payoff table. To study the effect of scaling the system to many agents, this methodology would greatly benefit from a systematic approximation of payoffs in truly large finite populations. This direction could provide the basis for determining how many agents are required to achieve a certain global system behavior, and how many are *probably sufficiently many agents* to yield a good approximation of real systems.

Despite the limitations that are bound to arise with any choices in the methodology, the sum of the parts forms a coherent extension of the framework for state-of-the-art multi-agent learning research. I hope to have given the reader a comprehensive overview of the research that has been performed, and leave it to the interested mind to ponder further promising research based on the potential but also limitations that have been pointed out throughout this dissertation.

Bibliography

Sherief Abdallah and Victor Lesser. A Multiagent Reinforcement Learning Algorithm with Non-linear Dynamics. *Journal of Artificial Intelligence Research*, 33:521–549, 2008. 4, 5, 21, 35, 85

Rajeev Agraval. Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Applied Probability*, 27(4):1054–1078, 1995. 13

James J. Angel. Market mechanics: A guide to U.S. stock markets. Technical report, The Nasdaq Stock Market Educational Foundation, 2002. 92, 93

Dan Ariely. *Predictably irrational: The hidden forces that shape our decisions.* Harper-Collins, 2009. 40

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best Arm Identification in Multi-Armed Bandits. In *Proc. of the 23rd Int. Conf. on Learning Theory (COLT)*, 2010. 13

Peter Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002. 4, 13

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 2002. 13

Monica Babes, Michael Wunder, and Michael Littman. Q-learning in two-player two-action games. In *Adaptive Learning Agents Workshop (ALA 2009)*, 2009. 39, 53

Patrick Bajari and Ali Hortacsu. The Winners Curse, Reserve Prices and Endogenous Entry: Empirical Insights From eBay Auctions. *RAND Journal of Economics*, 34(2): 329–355, 2003. 92, 93

Dave Barnes, Andy Shaw, and Steve Pugh. Autonomous Sample Acquisition for the ExoMars Rover. In *Proc. of the 9th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, Noordwijk, The Netherlands, 2006. 2

Christian Barrot, Sönke Albers, Bernd Skiera, and Björn Schäfers. 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, July 2010. 93

Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. 15

Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating Game-Theoretic Optimal Strategies for Full-scale Poker. In *Proc. of 18th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 661–668, 2003. 104

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. A comparative study of multi-agent reinforcement learning dynamics. In *Proc. of 22nd Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC 2010)*, pages 11–18. University of Luxembourg, 2010a. 69, 70

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Lenient frequency adjusted Q-learning. In *Proc. of 22nd Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC 2010)*, pages 19–26. University of Luxembourg, 2010b. 69, 70

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Empirical and Theoretical Support for Lenient Learning (Extended Abstract). In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1105–1106. International Foundation for AAMAS, 2011. 6, 55

Avrim Blum and Yishay Mansour. Learning, Regret minimization, and Equilibria. In Nisan, Roughgarden, Tardos, and Vazirani, editors, *Algorithmic Game Theory*, chapter 4, pages 79–100. Cambridge University Press, 2007. 4, 17, 32

Tilman Börgers and Rajiv Sarin. Learning Through Reinforcement and Replicator Dynamics. *Journal of Economic Theory*, 77(1):1–14, 1997. 5, 25, 30, 31, 34, 39, 46, 83, 85, 89

Graham Bowley. Lone $4.1 Billion Sale Led to Flash Crash in May. *The New York Times*, October 2, 2010. 3

Michael Bowling. Convergence and No-Regret in Multiagent Learning. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 209–216, 2005. 31

Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002. 5, 21, 31, 34, 38, 85

Ronen I. Brafman and Moshe Tennenholtz. R-max - A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. *Journal of Machine Learning Research*, 3:213–231, 2002. 69

Doyle Brunson. *Doyle Brunson's Super System: A Course in Power Poker*. Cardoza, 1979. 103, 105, 107

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 18(2):156–172, 2008. 3, 4, 16, 20, 37, 83, 118

Doran Chakraborty and Peter Stone. Convergence, Targeted Optimality, and Safety in Multiagent Learning. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proc. of the 27th Int. Conf. on Machine Learning (ICML)*, pages 191–198. Omnipress, 2010. 21

Howie Choset. Coverage for robotics - A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001. 2

Dave Cliff. Minimal-Intelligence Agents for Bargaining Behaviors in Market-Based Environments. Technical Report September 1996, Technical Report HP/97/91, Hewlett Packard Laboratories, Bristol, England, 1997. 62

Dave Cliff and Janet Bruten. Zero Not Enough: On The Lower Limit of Agent Intelligence For Continuous Double Auction Markets. Technical report, Technical Report HP/97/141, Hewlett Packard Laboratories, Bristol, England, March 1997. 62

Dave Cliff and Janet Bruten. Less than human: Simple adaptive trading agents for CDA markets. Technical report, Technical Report HP/97/155, Hewlett Packard Laboratories. To be presented at CEFEES98, Cambridge UK, Bristol, England, 1998. 62

Alfred Cowles. Can stock market forecasters forecast? *Econometrica: Journal of the Econometric Society*, pages 309–324, 1933. 92

Jacob W. Crandall. Just Add Pepper: Extending Learning Algorithms for Repeated Matrix Games to Repeated Markov Games. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 399–406. International Foundation for Autonomous Agents and Multiagent Systems, 2012. 69

Jacob W. Crandall, Asad Ahmed, and Michael A. Goodrich. Learning in Repeated Games with Minimal Information: The Effects of Learning Bias. In *Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, pages 650–656, 2011. 4, 16, 20, 21, 37, 40, 82

John G. Cross. A Stochastic Learning Model of Economic Behavior. *The Quarterly Journal of Economics*, 87(2):239–266, 1973. 3, 5, 16

Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. In *Proc. of the Int. Conf. on Artificial Intelligence (ICAI)*, pages 1467–1473, 2000. 103

Yann-Michaël De Hauwere, Peter Vrancx, and Ann Nowé. Solving Delayed Coordination Problems in MAS (Extended Abstract). In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1115–1116, 2011. 21

Bruce Bueno de Mesquita. Game Theory, Political Economy, and the Evolving Study of War and Peace. *American Political Science Review*, 100(4):637–642, 2006. 1

Department for Business Innovation and Skill. Foresight annual review 2011. Technical report, UK Department for Business Innovation and Skills, 2012. 3

Ido Erev and Alvin E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, pages 848–881, 1998. 62

Eyal Even-dar, Shie Mannor, and Yishay Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In *Proc. of 15th Annual Conf. on Computational Learning Theory (COLT)*, pages 255–270, 2002. 13

Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics including Feynman's Tips on Physics: The Definitive and Extended Edition.* Addison Wesley, 2005. 77

Justine Fox. *The Myth of the Rational Market: A History of Risk, Reward, and Delusion on Wall Street.* Harper Paperbacks, 2009. 102

Daniel Friedman. The double auction market institution: A survey. In Daniel Friedman and John Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*, volume 14, pages 3–25. Addison-Wesley, 1993. 93

Robert Gibbons. *A Primer in Game Theory.* Pearson Education, 1992. 4, 22

Herbert Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction.* University Press, Princeton, NJ, 2nd edition, 2009. 5, 25, 33, 85, 104

John Gittins, Richard Weber, and Kevin Glazebrook. *Multi-armed bandit allocation indices.* Wiley, 2nd edition, 2011. 13

Steven Gjerstad and John Dickhaut. Price Formation in Double Auctions. *Games and Economic Behavior*, 22(1):1–29, 1998. 62

Eduardo Rodrigues Gomes and R. Kowalczyk. Dynamic Analysis of Multiagent Q-learning with ϵ-greedy Exploration. In *Proc. of the 26th Annual Int. Conf. on Machine Learning (ICML 2009)*, pages 369–376, 2009. 39, 53, 66

Dan Harrington. *Harrington on Holdem Expert Strategy for No Limit Tournaments.* Two Plus Two Publisher, 2004. 103, 105, 107

Daniel Hennes, Daan Bloembergen, Michael Kaisers, Karl Tuyls, and Simon Parsons. Evolutionary Advantage of Foresight in Markets. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 943–949, 2012. 91

Daniel Hennes, Michael Kaisers, and Karl Tuyls. RESQ-learning in stochastic games. In *Adaptive and Learning Agents (ALA 2010) Workshop*, 2010. 55, 68, 69

Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos.* Academic Press, 2004. 24, 26, 47, 48

Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics.* Cambridge University Press, 2002. 24, 25, 27, 30, 32, 89, 104, 119

Shlomit Hon-Snir, Dov Monderer, and Aner Sela. A Learning Approach to Auctions. *Journal of Economic Theory*, 82:65–88, 1998. 1

Ronald A. Howard. *Dynamic Programming and Markov Process.* MIT Press, 1960. 14, 15

Junling Hu and Michael P Wellman. Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research*, 4:1039–1069, 2003. 4, 21

Jürgen Huber. “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, 2007. 92, 97, 102

Jürgen Huber, Michael Kirchler, and Matthias Sutter. Is more information always better? Experimental financial markets with cumulative information. *Journal of Economic Behavior and Organization*, 65(1):86–104, 2008. 92, 94, 97, 99, 102

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010. 13

Leslie Pack Kaelbling, Michael Lederman Littman, and Andrew William Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. 4, 13, 16, 20, 118

Michael Kaisers. Replicator Dynamics for Multi-agent Learning - An Orthogonal Approach. In Toon Calders, Karl Tuyls, and Mykola Pechenizkiy, editors, *Proc. of the 21st Benelux Conference on Artificial Intelligence (BNAIC 2009)*, pages 113–120, Eindhoven, 2009. 75

Michael Kaisers, Daan Bloembergen, and Karl Tuyls. A Common Gradient in Multi-agent Reinforcement Learning (Extended Abstract). In Conitzer, Winikoff, Padgham, and van der Hoek, editors, *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1393–1394. International Foundation for AAMAS, 2012. 75

Michael Kaisers and Karl Tuyls. Frequency Adjusted Multi-agent Q-learning. In van der Hoek, Kamina, Lespérance, Luck, and Sen, editors, *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315. International Foundation for AAMAS, 2010. 5, 38, 85

Michael Kaisers and Karl Tuyls. FAQ-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), 2011. 38, 53

Michael Kaisers and Karl Tuyls. Multi-agent Learning and the Reinforcement Gradient. In Massimo Cossentino, Michael Kaisers, Karl Tuyls, and Gerhard Weiss, editors, *Multi-Agent Systems. 9th European Workshop, EUMAS 2011*, pages 145–159. Lecture Notes in Computer Science, Vol. 7541. Springer, 2012. 75

Michael Kaisers, Karl Tuyls, and Simon Parsons. An Evolutionary Model of Multi-agent Learning with a Varying Exploration Rate (Extended Abstract). In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1255–1256. International Foundation for AAMAS, 2009. 55, 94

Michael Kaisers, Karl Tuyls, Frank Thuijsman, and Simon Parsons. Auction Analysis by Normal Form Game Approximation. In *Proc. of Int. Conf. on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008)*, pages 447–450. IEEE/WIC/ACM, December 2008. 62

Ardeshir Kianercy and Aram Galstyan. Dynamics of Boltzmann Q-Learning in Two-Player Two-Action Games. *Physics Review E*, 85(4), 2012. 53, 77, 89

Michael Kirchler. Partial knowledge is a dangerous thing - On the value of asymmetric fundamental information in asset markets. *Journal of Economic Psychology*, 31(4): 643–658, 2010. 92, 97, 98, 100, 102

Tomas Klos, Gerrit Jan van Ahee, and Karl Tuyls. Evolutionary Dynamics of Regret Minimization. In Balcázar, Bonchi, Gionis, and Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6322 of *Lecture Notes in Computer Science*, pages 82–96. Springer Berlin / Heidelberg, 2010. 5, 18, 32, 84, 85

Tom Lauriciella, Kara Scannell, and Jenny Strasburg. How a Trading Algorithm Went Awry. *The Wall Street Journal*, October 1, 2010. 3

Michael L. Littman. Friend-or-foe Q-learning in general-sum games. In *Proc. of the 18th Int. Conf. on Machine Learning (ICML)*, pages 322–328, 2001. 21

Michael Lederman Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of 11th Int. Conf. on Machine Learning (ICML)*, pages 157–163, 1994. 15, 21

Burton G. Malkiel. The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, 17(1):59–82, 2003. 92

John Maynard Smith. *Evolution and the Theory of Games.* Cambridge University Press, 1982. 104

John McMillan. Selling Spectrum Rights. *Journal of Economic Perspectives*, 8(3): 145–162, 1994. 93

Kumpati Subrahmanya Narendra and Mandayam A. L. Thathachar. Learning Automata - A Survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 4(4):323–334, 1974. 17

Abraham Neyman. From Markov Chains to Stochastic Games. In Abraham Neyman and Sylvain Sorin, editors, *Stochastic Games and Applications*, pages 397–415. Kluwer Academic Publishers, 2003. 15

James Nicolaisen, Valentin Petrov, and Leigh Tesfatsion. Market Power and Efficiency in a Computational Electricity Market With Discriminatory Double-Auction Pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523, 2001. 62

Jinzhong Niu, Kai Cai, Simon Parsons, Maria Fasli, and Xin Yao. A grey-box approach to automated mechanism design. *Electronic Commerce Research and Applications*, 11(1):24–35, January 2012. 93

Shervin Nouyan, Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo. Teamwork in Self-Organized Robot Colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711, 2009. 1

Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, November 2005. 2, 4, 20, 118

Liviu Panait, Karl Tuyls, and Sean Luke. Theoretical Advantages of Lenient Learners: An Evolutionary Game Theoretic Perspective. *Journal of Machine Learning Research*, 9:423–457, 2008. 69, 70, 73

Simon Parsons, Marek Marcinkiewicz, and Jinzhong Niu. Everything you wanted to know about double auctions, but were afraid to (bid or) ask. Technical report, Brooklyn College, City University of New York, 2005. 62

Simon Parsons, Juan A. Rodriguez-Aguilar, and Mark Klein. Auctions and Bidding: A Guide for Computer Scientists. *ACM Computing Surveys (CSUR)*, 43(2):10, 2011. 93

Steve Phelps, Marek Marcinkiewicz, and Simon Parsons. A novel method for automatic strategy acquisition in N-player non-zero-sum games. In *Proc. of the fifth Int. joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS '06)*, pages 705–712, New York, New York, USA, 2006. ACM Press. 1, 61, 93

Steve Phelps, Peter McBurney, and Simon Parsons. A Novel Method for Strategy Acquisition and its Application to a Double-Auction Market Game. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 40(3):668–674, 2010a. 61, 93

Steve Phelps, Peter McBurney, and Simon Parsons. Evolutionary mechanism design: a review. *Journal of Autonomous Agents and Multi-Agent Systems*, 21(2):237–264, October 2010b. 61, 93

Steve Phelps, Simon Parsons, and Peter McBurney. Automated trading agents verses virtual humans: An evolutionary game-theoretic comparison of two double-auction market designs. In *Proc. of the 6th Workshop on Agent-Mediated Electronic Commerce*, New York, USA, 2004. 61, 62, 104

Steve Phelps, Simon Parsons, and Peter Mcburney. An evolutionary game-theoretic comparison of two double-auction market designs. In *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems*, pages 101–114, 2005. 94

Marc Ponsen, Jan Ramon, Tom Croonenborghs, Kurt Driessens, and Karl Tuyls. Bayes-Relational Learning of Opponent Models from Incomplete Information in No-Limit Poker Learning an Opponent Model. In *Proc. of 23rd Conf. of the Association for the Advancement of Artificial Intelligence (AAAI-08)*, pages 1485–1487, Chicago, USA, 2008. 103

Marc Ponsen, Karl Tuyls, Michael Kaisers, and Jan Ramon. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1(1):39–45, January 2009. 91

Rob Powers and Yoav Shoham. New Criteria and a New Algorithm for Learning in Multi-Agent Systems. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 1089–1096, 2004. 21

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994. 14

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. 12

John Rust, Richard Palmer, and John H. Miller. Behavior of Trading Automata in a Computerized Double Auction Market. In *The Double Auction Market: Institutions, Theories, and Evidence*. Santa Fe Institute, Addison-Wesley, 1993. 62

William H. Sandholm. *Population Games and Evolutionary Dynamics*. MIT Press, 2010. 88

Thomas D. Schneider. Evolution of biological information. *Nucleic Acids Research*, 28 (14):2794–9, July 2000. 33

Lloyd Stowell Shapley. Stochastic games. *Proc. of the National Academy of Sciences*, 39:1095–1100, 1953. 16

Yoav Shoham and Kevin Leyton-brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009. 16

Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, May 2007. 2, 3, 4, 20, 37, 118

Satinder Singh, Michael Kearns, and Yishay Mansour. Nash Convergence of Gradient Dynamics in General-Sum Games 3 Gradient Ascent for Iterated Games. In *Proc. of 16th Conference on Uncertainty in Artificial Intelligence (UAI 2000)*, pages 541–548, Stanford, 2000. Morgan. 5, 31, 34, 85

David Sklansky. *The Theory of Poker By.* Two Plus Two Publisher, 1987. 103, 105, 106, 107

John Maynard Smith. *Evolution and the Theory of Games.* Cambridge University Press, 1982. 24

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes Bluff: Opponent Modelling in Poker. In *Proc. of the 21st Conf. in Uncertainty in Artificial Intelligence (UAI)*, pages 550–558, 2005. 103

Dietrich Stauffer. Life, Love and Death: Models of Biological Reproduction and Aging. Technical report, Institute for Theoretical Physics, 1999. 33

Peter Stone. Multiagent learning is not the answer. It is the question. *Artificial Intelligence*, 171(7):402–405, May 2007. 3, 4

Peter Stone and Manuela Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robotics*, 8(3):345–383, 2000. 2

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. *Proceedings of the 23rd Int. Conf. on Machine learning (ICML)*, pages 881–888, 2006. 13

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge MA, 1998. 2, 5, 11, 12, 15, 18, 19, 33, 34, 37, 104

Peter D. Taylor and Leo B. Jonker. Evolutionary Stable Strategies and Game Dynamics. *Mathematical Biosciences*, 156:145–156, 1978. 24, 104, 106

Mandayam A. L. Thathachar and P. S. Sastry. Varieties of Learning Automata: An Overview. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 32(6):711–722, 2002. 17

Mandayam A. L. Thathachar and P. S. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization.* Springer, 2003. 32

Bence Tóth and Enrico Scalas. The value of information in financial markets: An agent-based simulation. In Jürgen Huber and Michael Hanke, editors, *Information, Interaction, and (In)Efficiency in Financial Markets*, pages 1–25. Linde Verlag, 2007. 92, 94, 96, 97, 98, 100, 102

Bence Tóth, Enrico Scalas, Jürgen Huber, and Michael Kirchler. Agent-based simulation of a double-auction market with heterogeneously informed agents. In *Potentials of Complexity Science for Business, Governments, and the Media*, 2006. 92, 96, 97, 102

Bence Tóth, Enrico Scalas, Jürgen Huber, and Michael Kirchler. The value of information in a multi-agent market model. *The European Physical Journal B*, 55(1): 115–120, February 2007. 92, 94, 97, 99, 102

John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3):185–202, September 1994. 59

Karl Tuyls and Simon Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):406–416, May 2007. 3, 5, 25, 30

Karl Tuyls, Pieter Jan 't Hoen, and Bram Vanschoenwinkel. An Evolutionary Dynamical Analysis of Multi-Agent Learning in Iterated Games. *Journal of Autonomous Agents and Multi-Agent Systems*, 12(1):115–153, 2006. 5, 6, 19, 25, 33, 40, 41, 42, 46, 53, 67, 83, 104, 106

Karl Tuyls, Katja Verbeeck, and Tom Lenaerts. A Selection-Mutation model for Q-learning in Multi-Agent Systems. In *Proceedings of AAMAS 2003, The ACM International Conference Proceedings Series*, 2003. 5, 19, 33, 40, 41, 42, 46, 53, 66, 70, 106

Leslie G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984. 13

Jaap H. van den Herik, Daniel Hennes, Michael Kaisers, Karl Tuyls, and Katja Verbeeck. Multi-agent learning dynamics: A survey. In *Cooperative Information Agents XI, LNAI*, volume 4676, pages 36–56. Springer, 2007. 4

Peter Vrancx, Katja Verbeeck, and Ann Nowé. Networks of learning automata and limiting games. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pages 224–238. Springer, 2008. 68, 69

Perukrishnen Vytelingum, Dave Cliff, and Nicholas R. Jennings. Analysing Buyers and Sellers Strategic Interactions in Marketplaces: An Evolutionary Game Theoretic Approach. In *Proc. of the 9th Int. Workshop on Agent-Mediated Electronic Commerce (AMEC)*, Hawaii, USA, 2007. 104

W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing Complex Strategic Interactions in Multi-Agent Systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, pages 109–118, 2002. 6, 29, 62, 91, 104

Christopher J. C. H. Watkins and Peter Dayan. Q-Learning. *Machine Learning*, 8: 279–292, 1992. 4, 5, 6, 18, 37, 40, 55

Richard Weber. On the gittins index for multiarmed bandits. *Annals of Applied Probability*, 2(4):1024–1033, 1992. 13

Jörgen W. Weibull. *Evolutionary Game Theory*. MIT Press, 1996. 25, 33

Gerhard Weiß. Distributed reinforcement learning. *Robotics and Autonomous Systems*, 15:135–142, 1995. 20

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, May 1992. 31, 89

Michael Wunder, Michael Littman, and Monica Babes. Classes of Multiagent Q-learning Dynamics with ϵ-greedy Exploration. In *Proc. of the 27th Int. Conf. on Machine Learning*, pages 1167–1174, Haifa, Israel, 2010. Omnipress. 5, 27, 39, 53, 66

Erik Christopher Zeeman. Dynamics of the Evolution of Animal Conflicts. *Journal of Theoretical Biology*, 89(2):249–270, 1981. 106

Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proc. of the 20th Int. Conf. on Machine Learning (ICML)*, pages 421–422, 2003. 31

List of figures

List of tables

Notation

α learning rate α

β auxiliary FAQ-learning rate β

(A, B) Payoff bi-matrix, with A containing payoffs for the row player, and B containing payoffs for the column player

C_{ij} payoff matrix C, where C_{ij} indicates the payoff of action i against j for the column player

δ the small positive constant δ can usually be chosen arbitrarily small, used as a continuous time increment, or to denote *in probability* as $1 - \delta$

∂, d differential in a continuous time process

Δ difference in a discrete time process

e_i the i^{th} unit vector e_i, a vector with all zeros except the i^{th} component which is one

ϵ error ϵ is a small positive value, used to bound the difference between actual and optimal behavior or performance; in the market model it denotes a small random variable

γ discount factor γ, used in the discounted sum of future returns

$\eth$ gradient of the value or payoff function

h the vector $h = (1, -1)$

i action i, usually used as an iterator or arbitrary action

j action j denotes a specific action, or used as an auxiliary iterator

$J(\cdot, \cdot)$ Jacobian $J(x, y)$ of a dynamical system based on the joint policy state (x, y)

k finite number of actions k

λ eigenvalue λ

n finite total number of elements n

p_s probability p_s of being in state s under the current policy

$\pi(s)$ deterministic policy, where $\pi(s)$ returns the action played in state s

$Q_i(s,t)$ Q-value at time t for taking action i in state s

$r(t)$ reward $r(t)$ at time t

$R_i(s,s')$ reward function $R_i(s,s')$ in a Markov decision process, given after selecting action i in state s and moving the process into state s'

R_{ij} payoff matrix R, where R_{ij} indicates the payoff of action i against j for the row player

s discrete state s of a Markov decision process or stochastic game

t time t; in algorithms discrete, i.e., $t \in \mathbb{N}$, in dynamical systems continuous, i.e., $t \in \mathbb{R}$

$T_i(s,s')$ transition function $T_i(s,s')$, denoting the probability to move to state s' after selection action i in state s

τ temperature τ, exploration rate, either considered constant or a function of time

u the uniform policy $u = (\frac{1}{n}, \ldots, \frac{1}{n})$, assuming n actions

$V(x,y)$ value function $V(x,y)$

w regret minimization weights of the polynomial weights algorithm

x policy of the row player $x = (x_1, \ldots, x_k)$, with x_i denoting the probability for playing action i

X_k the $(k-1)$-dimensional simplex over k actions

$\chi(x_0,t)$ the trajectory point that is reached from initial policy x_0 by following the dynamics $\frac{dx}{dt}$ for t units of continuous time

x^e Nash equilibrium policy of the row player

y policy of the column player

Summary

Computer programs automate increasingly complex tasks. Previously, tasks could be predefined, e.g., for industrial robotics. In contrast, new application domains like automated stock trading require highly adaptive agents that learn in dynamic environments and against adversarial opponents. While automated trading is increasingly adopted and now generates about a third of all trading volume in the UK, the understanding of systems in which agents are *learning against learning* is limited. The lack of a formal framework makes assessing the stability of these crucial systems practically impossible. This dissertation addresses the need for a formal framework to analyze multi-agent learning, drawing on an established relationship between multi-agent reinforcement learning and evolutionary game theory.

Previous work has shown that the behavior of stochastic multi-agent learning algorithms with an infinitesimal learning rate can be described by deterministic dynamical systems. This approach makes it possible to employ tools from dynamical systems theory to judge the convergence properties of learning algorithms in strategic interactions. In particular, the dynamics of Q-learning have been related to an extension of the replicator dynamics from evolutionary game theory with an additional exploration term. However, this equivalence is based on the simplifying assumption that all actions are updated at every time step. Here, I show that this leads to a discrepancy between the observed algorithm performance and the idealized evolutionary model. Since the idealized model shows preferable behavior, I introduce the variation *Frequency Adjusted Q-learning* (FAQ-learning) that adheres to the idealized dynamics. In addition, this solidified link is used to provide a convergence proof for FAQ-learning in two-agent two-action games. In the limit of infinite time, FAQ-learning converges to stable points whose distance to Nash equilibria is related to the degree of exploration of the algorithms. Hence, this proof relates multi-agent reinforcement learning to evolutionary and classical game theory.

In subsequent chapters, I extend the evolutionary framework for multi-agent learning to more realistic settings, like multiple states and varying exploration rates. Furthermore, I introduce an orthogonal visualization of the dynamical systems that provides a method to design time-dependent parameters of agents (e.g., exploration) and games. The evolutionary game theoretic models have the replicator dynamics as a common building block, and a similar term appears in the dynamical systems describing *Infinitesimal Gradient Ascent* (IGA). The commonalities and differences between variations of IGA dynamics and replicator dynamics are discussed in detail. In essence, the difference depends on whether the payoff signal is known for all actions at every time step or whether it needs to be sampled for one action at a time. This implies that the reinforcement-learning algorithms can be seen as stochastic gradient ascent on the payoff function. The comparative discussion of these two independently developed approaches unites them under the same terminology and provides a basis for further cross-fertilization.

Finally, the merits of an evolutionary analysis are demonstrated in two application domains: auctions and poker. The analysis critically evaluates strategic behavior and compares the results with domain knowledge. The strategic payoffs from the application domains are captured in a *heuristic payoff table* by observing various finite strategy constellations. Subsequently, the expected payoff for an arbitrary mix of strategies in an infinite population can be approximated from the heuristic payoff table, and is used in the context of the evolutionary dynamics. In poker, results are in line with expert advice, even more so if exploration is accounted for in the evolutionary model. Similarly, results in simulated double auctions confirm results from previous work. More specifically, performance in double auctions does not increase monotonically with more information about the future price development: traders with no information perform at market average, while traders with little information are exploited by insiders with a lot of information; this results in a J-curve for the value of information. If information comes for free, insiders drive other traders extinct. If on the other hand information is costly, less informed traders may prevail. This work provides a good basis to study the resilience to exogenous events, like trader in- and outflow, that may further disturb the system.

Overall, this dissertation contributes to the state-of-the-art in multi-agent reinforcement learning in several ways: (1) a critical evaluation and improvement of the link between Q-learning and its idealized dynamics enables a proof of convergence for the variant Frequency Adjusted Q-learning, (2) the evolutionary framework is extended to more realistic settings and enriched by new perspectives, and (3) application domains demonstrate how practical insights can be derived from the theoretical models. Tying together tools from reinforcement learning, dynamical systems, evolutionary and classical game theory, this dissertation lays out a formal framework for the analysis of systems in which agents are learning against learning, paving the way for many viable future research endeavors.

Samenvatting

In dit proefschrift bestudeer ik computerprogramma's (agenten) die samen leren te coördineren of te concurreren. Er wordt hoofdzakelijk onderzocht hoe hun leerprocessen elkaar beïnvloeden. Dergelijke adaptieve agenten spelen reeds een belangrijke rol in onze maatschappij. Zo nemen geautomatiseerde agenten bijvoorbeeld al deel aan de financiële handel en genereren in een aantal Amerikaanse markten reeds meer transacties dan de mens. Ondanks de grootschalige toepassing is het voor de meerderheid van leeralgoritmen enkel bewezen dat ze goed presteren als zij geïsoleerd optreden—zodra een tweede agent invloed heeft op de omgeving of uitkomsten, zijn de meeste garanties niet meer van toepassing. Mijn belangrijkste bijdragen zijn de uitbreiding en de toepassing van methodiek om te beoordelen in hoeverre optimaal gedrag in strategische interacties door leeralgoritmen wordt benaderd. Het gedrag van deze algoritmen wordt geformaliseerd op basis van stochastische en dynamische systemen, en hun korte en lange termijn prestaties worden in het kader van de klassieke en evolutionaire speltheorie besproken.

Zusammenfassung

In dieser Dissertation werden Computerprogramme (Agenten) analysiert, die mit- und gegeneinander lernen. Im Besonderen wird darauf eingegangen, wie sich die Lernprozesse der Agenten gegenseitig beeinflussen. Solche adaptive Agenten spielen schon heute in verschiedenen Bereichen unseres Lebens eine ausschlaggebende Rolle, auch wenn dies häufig übersehen wird; so nehmen z.B. Computer-Agenten an Finanzmärkten teil und generieren in einigen US Märkten größere Transaktionsvolumen als menschliche Händler. Für allein agierende lernende Agenten bzw. deren Lernverfahren können Konvergenz zum optimalen Verhalten und dessen Stabilität häufig garantiert werden. Solche Garantien sind für Systeme, bestehend aus mehreren, interagierenden, lernenden Agenten, im Allgemeinen nicht übertragbar, da das optimale Verhalten (das Lernziel) des einen Agenten vom Verhalten der anderen Agenten abhängt und sich fortwährend ändern kann. In der vorliegenden Dissertation wird eine Methode entwickelt und angewandt, die es erlaubt zu bewerten, inwiefern sich interagierende Lernverfahren an das theoretisch erreichbare Optimalverhalten in strategischen Konflikten annähern. Das Verhalten dieser Lernverfahren wird mit Hilfe von stochastischen und dynamischen Systemen formal modelliert, und das Kurz- und Langzeitverhalten wird im Kontext von klassischen und evolutionären spieltheoretischen Lösungansätzen diskutiert.

List of publications[1]

Haitham Bou Ammar, Karl Tuyls, and Michael Kaisers. Evolutionary Dynamics of Ant Colony Optimization. In Ingo J. Timm and Christian Guttmann, editors, *Multiagent System Technologies. 10th German Conference, MATES 2012*, pages 40–52. Lecture Notes in Computer Science, Vol. 7598. Springer, 2012.

Daniel Hennes, Daan Bloembergen, Michael Kaisers, Karl Tuyls, and Simon Parsons. Evolutionary Advantage of Foresight in Markets. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 943–949, 2012. 91

Michael Kaisers, Daan Bloembergen, and Karl Tuyls. A Common Gradient in Multi-agent Reinforcement Learning (Extended Abstract). In Conitzer, Winikoff, Padgham, and van der Hoek, editors, *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1393–1394. International Foundation for AAMAS, 2012. 75

Michael Kaisers and Karl Tuyls. Multi-agent Learning and the Reinforcement Gradient. In Massimo Cossentino, Michael Kaisers, Karl Tuyls, and Gerhard Weiss, editors, *Multi-Agent Systems. 9th European Workshop, EUMAS 2011*, pages 145–159. Lecture Notes in Computer Science, Vol. 7541. Springer, 2012. 75

Marcel Neumann, Karl Tuyls, and Michael Kaisers. Using Time as a Strategic Element in Continuous Double Auctions. In Ingo J. Timm and Christian Guttmann, editors, *Multiagent System Technologies. 10th German Conference, MATES 2012*, pages 106–115. Lecture Notes in Computer Science, Vol. 7598. Springer, 2012.

Michael Wunder, Michael Kaisers, John Robert Yaros, and Michael Littman. A Framework for Modeling Population Strategies by Depth of Reasoning. In Conitzer, Winikoff, Padgham, and van der Hoek, editors, *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 947–954. International Foundation for AAMAS, 2012.

Sjriek Alers, Daan Bloembergen, Daniel Hennes, Steven de Jong, Michael Kaisers, Nyree Lemmens, Karl Tuyls, and Gerhard Weiss. Bee-inspired foraging in an embodied swarm (Demonstration). In Tumer, Yolum, Sonenberg, and Stone, editors,

[1] Within the domain of computer science, high impact conferences such as AAMAS or AAAI are regarded comparable if not preferable to journals. For a scientific discussion see Research Evaluation for Computer Science by Bertrand Meyer et al. in Communications of the ACM.

Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011), pages 1311–1312. International Foundation for AAMAS, 2011.

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Empirical and Theoretical Support for Lenient Learning (Extended Abstract). In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1105–1106. International Foundation for AAMAS, 2011. 6, 55

Michael Kaisers and Karl Tuyls. FAQ-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), 2011. 38, 53

Daniel Mescheder, Karl Tuyls, and Michael Kaisers. Opponent Modeling with POMDPs. In *Proc. of 23nd Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC 2011)*, pages 152–159. KAHO Sint-Lieven, Gent, 2011.

Michael Wunder, Michael Kaisers, J.R. Yaros, and Michael Littman. Using iterated reasoning to predict opponent strategies. In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 593–600. International Foundation for AAMAS, 2011.

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. A comparative study of multi-agent reinforcement learning dynamics. In *Proc. of 22nd Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC 2010)*, pages 11–18. University of Luxembourg, 2010a. 69, 70

Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Lenient frequency adjusted Q-learning. In *Proc. of 22nd Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC 2010)*, pages 19–26. University of Luxembourg, 2010b. 69, 70

Daniel Hennes, Michael Kaisers, and Karl Tuyls. RESQ-learning in stochastic games. In *Adaptive and Learning Agents (ALA 2010) Workshop*, 2010. 55, 68, 69

Michael Kaisers and Karl Tuyls. Frequency Adjusted Multi-agent Q-learning. In van der Hoek, Kamina, Lespérance, Luck, and Sen, editors, *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315. International Foundation for AAMAS, 2010a. 5, 38, 85

Michael Kaisers and Karl Tuyls. Replicator Dynamics for Multi-agent Learning - An Orthogonal Approach. In Matthew E. Taylor and Karl Tuyls, editors, *Adaptive and Learning Agents. Second Workshop, ALA 2009*, pages 49–59. Lecture Notes in Computer Science, Vol. 5924. Springer, 2010b.

Michael Wunder, Michael Kaisers, Michael Littman, and John Robert Yaros. A Cognitive Hierarchy Model Applied to the Lemonade Game. In *Workshop on Interactive Decision Theory and Game Theory (IDTGT 2010)*. Assoc. for the Advancement of Artif. Intel. (AAAI), 2010.

Michael Kaisers. Replicator Dynamics for Multi-agent Learning - An Orthogonal Approach. In Toon Calders, Karl Tuyls, and Mykola Pechenizkiy, editors, *Proc. of the 21st Benelux Conference on Artificial Intelligence (BNAIC 2009)*, pages 113–120, Eindhoven, 2009. 75

Michael Kaisers, Karl Tuyls, and Simon Parsons. An Evolutionary Model of Multi-agent Learning with a Varying Exploration Rate (Extended Abstract). In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1255–1256. International Foundation for AAMAS, 2009. 55, 94

Marc Ponsen, Karl Tuyls, Michael Kaisers, and Jan Ramon. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1(1):39–45, January 2009. 91

Michael Kaisers, Karl Tuyls, and Frank Thuijsman. Discovering the game in auctions. In *Proc. of 20th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC 2008)*, pages 113–120. University of Twente, 2008a.

Michael Kaisers, Karl Tuyls, Frank Thuijsman, and Simon Parsons. Auction Analysis by Normal Form Game Approximation. In *Proc. of Int. Conf. on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008)*, pages 447–450. IEEE/WIC/ACM, December 2008b. 62

Jaap H. van den Herik, Daniel Hennes, Michael Kaisers, Karl Tuyls, and Katja Verbeeck. Multi-agent learning dynamics: A survey. In *Cooperative Information Agents XI, LNAI*, volume 4676, pages 36–56. Springer, 2007. 4

SIKS dissertation series

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*

2 Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*

3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*

4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*

10 Jan Wielemaker (UVA) *Logic programming for knowledge-intensive interactive applications*

11 Alexander Boer (UVA) *Legal Theory, Sources of Law & the Semantic Web*

12 Peter Massuthe (TUE) *Humboldt-Universitaet Berlin*Operating Guidelines for Services

13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*

14 Maksym Korotkiy (VU) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

15 Rinke Hoekstra (UVA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*

16 Fritz Reul (UvT) *New Architectures in Computer Chess*

17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

18 Fabian Groffen (CWI) *Armada, An Evolving Database System*

19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20 Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*

22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*

23 Peter Hofgesang (VU) *Modelling Web Usage in a Changing Environment*

24 Annerieke Heuvelink (VU) *Cognitive Models for Training Simulations*

25 Alex van Ballegooij (CWI) *RAM: Array Database Management through Relational Mapping*

26 Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*

27 Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*

28 Sander Evers (UT) *Sensor Data Management with Probabilistic Models*

29 Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*

30 Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*

31 Sofiya Katrenko (UVA) *A Closer Look at Learning Relations from Text*

32 Rik Farenhorst and Remco de Boer (VU) *Architectural Knowledge Management: Supporting Architects and Auditors*

33 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*

34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*

35 Wouter Koelewijn (UL) *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*

36 Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*

37 Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*

38 Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution – A Behavioral Approach Based on Petri Nets*

40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*

41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*

42 Toine Bogers (UvT) *Recommender Systems for Social Bookmarking*

43 Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

44 Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*

45 Jilles Vreeken (UU) *Making Pattern Mining Useful*

46 Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1 Matthijs van Leeuwen (UU) *Patterns that Matter*

2 Ingo Wassink (UT) *Work flows in Life Science*

3 Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*

4 Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

5 Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*

6 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*

7 Wim Fikkert (UT) *Gesture interaction at a Distance*

8 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

9 Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*

10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*

11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*

12 Susan van den Braak (UU) *Sensemaking software for crime analysis*

13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*

14 Sander van Splunter (VU) *Automated Web Service Reconfiguration*

15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*

16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*

17 Spyros Kotoulas (VU) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

18 Charlotte Gerritsen (VU) *Caught in the Act: Investigating Crime by Agent-Based Simulation*

19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*

20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*

22 Michiel Hildebrand (CWI) *End-user Support for Access to Heterogeneous Linked Data*

23 Bas Steunebrink (UU) *The Logical Structure of Emotions*

24 Dmytro Tykhonov (TUD) *Designing Generic and Efficient Negotiation Strategies*

25 Zulfiqar Ali Memon (VU) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

26 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

27 Marten Voulon (UL) *Automatisch contracteren*

28 Arne Koopman (UU) *Characteristic Relational Patterns*

29 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

30 Marieke van Erp (UvT) *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*

31 Victor de Boer (UVA) *Ontology Enrichment from Heterogeneous Sources on the Web*

32 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

33 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

34 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*

35 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*

36 Jose Janssen (OU) *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*

37 Niels Lohmann (TUE) *Correctness of services and their composition*

38 Dirk Fahland (TUE) *From Scenarios to components*

39 Ghazanfar Farooq Siddiqui (VU) *Integrative modeling of emotions in virtual agents*

40 Mark van Assem (VU) *Converting and Integrating Vocabularies for the Semantic Web*

41 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*

42 Sybren de Kinderen (VU) *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*

43 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

44 Pieter Bellekens (TUE) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

45 Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*

46 Vincent Pijpers (VU) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

47 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*

48 Jahn-Takeshi Saito (UM) *Solving difficult game positions*

49 Bouke Huurnink (UVA) *Search in Audiovisual Broadcast Archives*

50 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*

51 Peter-Paul van Maanen (VU) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

52 Edgar Meij (UVA) *Combining Concepts and Language Models for Information Access*

2011

1 Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2 Nick Tinnemeier (UU) *Work flows in Life Science*

3 Jan Martijn van der Werf (TUE) *Compositional Design and Verification of Component-Based Information Systems*

4 Hado van Hasselt (UU) *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms*

5 Base van der Raadt (VU) *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.*

6 Yiwen Wang (TUE) *Semantically-Enhanced Recommendations in Cultural Heritage*

7 Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*

8 Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*

9 Tim de Jong (OU) *Contextualised Mobile Media for Learning*

10 Bart Bogaert (UvT) *Cloud Content Contention*

11 Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*

12 Carmen Bratosin (TUE) *Grid Architecture for Distributed Process Mining*

13 Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UVA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Marc Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind's Eye on Personal Profiles*

20 Qing Gu (VU) *Guiding service-oriented software engineering - A view-based approach*

21 Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*

22 Junte Zhang (UVA) *System Evaluation of Archival Description and Access*

23 Wouter Weerkamp (UVA) *Finding People and their Utterances in Social Media*

24 Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

25 Syed Waqar ul Qounain Jaffry (VU) *Analysis and Validation of Models for Trust Dynamics*

26 Matthijs Aart Pontier (VU) *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

27 Aniel Bhulai (VU) *Dynamic website optimization through autonomous management of design patterns*

28 Rianne Kaptein (UVA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

29 Faisal Kamiran (TUE) *Discrimination-aware Classification*

30 Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*

31 Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

32 Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*

33 Tom van der Weide (UU) *Arguing to Motivate Decisions*

34 Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*

35 Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*

36 Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*

37 Adriana Birlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*

38 Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*

39 Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*

40 Viktor Clerc (VU) *Architectural Knowledge Management in Global Software Development*

41 Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*

42 Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*

43 Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*

44 Boris Reuderink (UT) *Robust Brain-Computer Interfaces*

45 Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*

46 Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*

47 Azizi Bin Ab Aziz (VU) *Exploring Computational Models for Intelligent Support of Persons with Depression*

48 Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*

49 Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1 Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*

2 Muhammad Umair (VU) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*

3 Adam Vanya (VU) *Supporting Architecture Evolution by Mining Software Repositories*

4 Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*

5 Marijn Plomp (UU) *Maturing Interorganisational Information Systems*

6 Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*

7 Rianne van Lambalgen (VU) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*

8 Gerben de Vries (UVA) *Kernel Methods for Vessel Trajectories*

9 Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*

10 David Smits (TUE) *Towards a Generic Distributed Adaptive Hypermedia Environment*

11 J.C.B. Rantham Prabhakara (TUE) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*

12 Kees van der Sluijs (TUE) *Model Driven Design and Data Integration in Semantic Web Information Systems*

13 Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*

14 Evgeny Knutov (TUE) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

15 Natalie van der Wal (VU) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.*

16 Fiemke Both (VU) *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*

17 Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*

18 Eltjo Poort (VU) *Improving Solution Architecting Practices*

19 Helen Schonenberg (TUE) *What's Next? Operational Support for Business Process Execution*

20 Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*

21 Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*

22 Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*

23 Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*

24 Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*

25 Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*

26 Emile de Maat (UVA) *Making Sense of Legal Text*

27 Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*

28 Nancy Pascall (UvT) *Engendering Technology Empowering Women*

29 Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*

30 Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*

31 Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

32 Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*

33 Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*

34 Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*

35 Evert Haasdijk (VU) *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*

36 Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*

37 Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*

38 Selmar Smit (VU) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*

39 Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*

40 Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*

41 Sebastian Kelle (OU) *Game Design Patterns for Learning*

42 Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*

43 (Withdrawn)

44 Anna Tordai (VU) *On Combining Alignment Techniques*

45 Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*

46 Simon Carter (UVA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

47 Manos Tsagkias (UVA) *A Model and Language for Business-aware Transactions*

48 Jorn Bakker (TUE) *Handling Abrupt Changes in Evolving Time-series Data*

49 Michael Kaisers (UM) *Learning against Learning - Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions*

"In this dissertation I study computer programs (agents) that learn to coordinate or to compete and investigate how their learning processes influence each other. Such adaptive agents already take vital roles behind the scenes of our society, e.g., automated agents participate in high frequency financial trading and create more transactions than human traders in some US markets. Despite their widespread application, many machine learning algorithms only have proven performance guarantees if they act alone; as soon as a second agent influences the outcomes most guarantees are invalid. My main contributions are the extension and application of the methodology to assess how closely algorithms approximate optimal behavior in strategic interactions. The behavior of these algorithms is formalized using models of stochastic and dynamical systems, and their short and long-term performance is discussed in terms of classical and evolutionary game theoretic solution concepts."

Michael Kaisers graduated from Maastricht University with a BSc in Knowledge Engineering in 2007 and a MSc in Artificial Intelligence in 2008. He earned the honor summa cum laude in both cases, while abbreviating the three-years bachelor's program to two years and complementing his master's program by an extra-curricular four-month research visit to Prof. dr. Simon Parsons at Brooklyn College, City University of New York.

In a nationwide competition, the Netherlands Organisation for Scientific Research (NWO) awarded him a TopTalent 2008 grant for his PhD research. In September 2008, he commenced his PhD position at Eindhoven University of Technology. From August 2009, the project continued at Maastricht University. He intensified his international research experience through a three-month research visit to Prof. dr. Michael Littman at Rutgers, State University of New Jersey, and published at various peer-reviewed workshops, conferences and journals. This dissertation coherently summarizes his PhD research.

ISBN 978-94-6169-331-0

9 789461 693310