

Game Theory and Agents

Stefan J. Johansson

Licentiate thesis submitted in partial fulfillment of the requirements for the degree of Licentiate in Computer Science at the University of Karlskrona/Ronneby.

Department of Software Engineering & Computer
Science,
University of Karlskrona/Ronneby

© October 25, 1999 Stefan J. Johansson

Department of Software Engineering and Computer Science
University of Karlskrona/Ronneby
Soft Center
S-372 25 Ronneby
Sweden

`stefan.johansson@ipd.hk-r.se`

`http://www.ipd.hk-r.se/~sja/`

ISBN 91-628-3907-1

Set in \LaTeX

Printed in Sweden by Kaserntryckeriet AB, Karlskrona

to my grandmothers

Abstract

A fundamental problem in multi agent systems is conflict resolution. A conflict occurs in general when the agents have to deal with conflicting goals, such as demands for shared, but limited resources. We investigate how game theory may be a helpful and efficient tool for examining a class of conflicts in multi agent systems and argue that parameterized games and conclusions based on general properties of strategies are of interest. The thesis includes four papers, the former three describing various parameterized simulations, and the latter one describing the theory behind it and stating some remarks given by our approach.

The first paper describes a parameterized hawk-and-dove game simulation that spans from the Chicken Game (CG) to the Prisoner's Dilemma (PD). We conclude that the CG to a higher extent rewards cooperative strategies than PD, and that the change in score is linear to the change of the payoff matrix.

In the second paper, we introduce the notion of generous and greedy strategies as being the ones that, given a certain distribution of opponents, play more (C,D) than (D,C) and more (D,C) than (C,D) respectively ((X,Y) is the move in which the opponent acts Y, when the own player acts X). The result, when simulated in CG and PD is that strategies starting as generous will outperform the greedy ones in a CG with a population game, while the opposite happens in the PD.

The last of the simulation based papers, compare four different PD like games and states how the generous and greedy strategies manage in the different games.

The fourth paper discuss why iterated games are interesting for multi agent research and how it is possible to use a pragmatic point of view in order to make an agent decide which strategy (or behavioral pattern) to use. The notion of *Characteristic Distributions (ChDs)* are introduced and it is used to support the agents in their choice. Based on the ChDs a "No Free Lunch" theorem for game theory is formulated and proved.

Preface

In the process of writing a thesis, a paper, or (as in this very moment) a section, there are two things that strikes at least me.

Firstly, how hard it is to get started and write the first sentences. One can come to think of a thousand *other* things to do; things that match the size of what you are about to write. *Aren't there any interesting courses that I could develop?* Of course there are! *Doesn't the simulation tool need some new features?* Naturally! Why not implement a multimedia applet that shows all results in runtime? *Isn't that L^AT_EX package beautiful?* You bet! Let's get right down to how it works in detail.

My solution to this problem of distractions have been to concentrate my time of writing to periods where I have no teaching, to work with people that have a subduing effect on my straggly way of working, and lately, to sit at home writing on my thesis.

The second thing that strikes me is how easy it is to write, once the steerage-way of the schooner of research is high enough. New areas reveal themselves as this cruise continues and I hope (and will work on towards) that these first chapters will give me enough wind to end up in an even more interesting waters.

Acknowledgments

The author wishes to express sincere appreciation to the University of Karlskrona/Ronneby, who have hosted me during my work with this thesis and especially the following four people (in no special order):

- PhLic. Bengt Carlsson¹, co-writer of three of the chapters (2–4) is a very valuable discussion partner. I think that we both have gained a lot on acting C.
- Professor Rune Gustavsson¹, who have granted my work and been my primary supervisor, especially during my first years, and who now is my main supervisor.
- PhD Paul Davidsson¹ has been a valuable support, first as a supplementary supervisor, and during the work of last year, especially on chapter 5, as my primary supervisor.
- Assistant professor Magnus Boman², co-writer of the papers in chapter 3 and 4 and reviewer of chapter 5 has also been to much help as a supplementary supervisor.

My (in many ways) remote supervisor during my time as a PhD student in Linköping, Patrick Doherty³ is worth to mention and I am very grateful to Lillemor Wallgren³ and her concerns about the status of my work, and ever lasting cheerful spirits.

Thanks also to my colleagues (present and former) in Societies of Computation, our research group: Fredrik Ygge⁴, whose comments on chapter 5 have been

¹Department of Computer Science, University of Karlskrona/Ronneby, Ronneby, Sweden

²Department of Computing and Systems Sciences, Stockholm University, Royal School of Technology, Stockholm, Sweden

³Department of Computer and Information Sciences, Linköping University, Linköping, Sweden

⁴Enersearch AB, Göteborg, Sweden

to helpful, Erik Astor⁵, Erik Persson⁵, Bertil Ekdahl⁵, Staffan Hägg⁶, Jens-Olof Lindh¹, Martin Fredriksson¹, Christer Rindebäck¹, Olle Lindeberg¹ and last but not least Marie Milette¹.

I would also like to thank Michael Mattsson¹ for his comments on an early draft of chapter 2, Gil Tidhar⁷ for his comments on the version presented at the DAI'97 workshop, Wojtek Kowalczyk⁸ for being an inspiration to start these studies from the beginning. Andreas Magnusson⁹ has given me well needed pieces of advice concerning the layout and Mikael Svanberg¹ for his help converting figure 2.2 from word to eps (one of the greater challenges in the work with this thesis).

I must not forget all the nice and helpful people at the department, Jenny G., Marie P., Ingrid P.M., Conny S., Tore D., Anders O., Anders C., Björn M., Petter N., Martin H. and Cecilia J. and the library personnel Kent, Pirjo and Inger, who have helped me with some of the references. If I have forgotten someone, please let me know and I will make it up in the next thesis.

At last, but not least, I would like to thank my family for being so patient with me during the time of writing this thesis.

⁵Department of Computing and Numerical Analysis, Lund University, Lund, Sweden

⁶Active Computing AB, Sölvesborg, Sweden

⁷Department of Computer Science, University of Melbourne, Melbourne, Australia

⁸Institutie Wiskunde en Informatica, Vrije Universiteit, Amsterdam, the Netherlands

⁹Intenia Research & Development AB, Linköping, Sweden

Contents

1	Introduction	1
1.1	The society level of MAS	1
1.1.1	The coordination mechanism component	3
1.1.2	The context component	3
1.1.3	The agent component	4
1.2	Game Theory	5
1.2.1	Temporal Aspects of Games	5
1.2.2	Symmetry in Games	7
1.2.3	Information in Games	8
1.2.4	Repeated vs Iterated Games	9
1.2.5	Noisy Games	11
1.2.6	Situated Games	12
1.2.7	Evolutionary Game Theory	13
1.2.8	Meta-games	13
1.3	Applications of Game Theory	13
1.3.1	Game theory as a tool for modeling interactions between artificial agents	15
1.4	On the Scientific Method	17
1.5	Outline of the Thesis	18
1.5.1	An Iterated Hawk-and-Dove Game	18
1.5.2	Generous and Greedy Strategies	19
1.5.3	Modeling Strategies as Generous and Greedy in Prisoner's Dilemma like Games	20
1.5.4	Characteristic Distributions in Iterated Games	21
1.5.5	Conclusions	21
1.5.6	The Appendices	21
2	An Iterated Hawk-and-Dove Game	23
2.1	Background	23

2.2	Disposition	24
2.3	A Comparison Between Three Different Games	24
2.3.1	The Prisoner's Dilemma	24
2.3.2	The Chicken Game	25
2.3.3	The Hawk-and-Dove Game	25
2.4	Evolutionary and Iterated Games	27
2.5	Examples of a Resource Allocation Problem	28
2.6	A Tournament Comparing Prisoner's Dilemma and Chicken Game	30
2.7	Discussion	34
2.8	Conclusions and Future Work	36
3	Generous and Greedy Strategies	39
3.1	Background	39
3.2	Methodology	39
3.3	A Generous-and-Greedy Model for Strategies	40
3.4	A Traffic Intersection Example	42
3.5	Simulating the Traffic Intersection Example	43
3.6	A Population Tournament	47
3.7	Adding Noise	48
3.8	Conclusions	50
4	Modelling Generosity and Greediness in Games	51
4.1	Introduction	51
4.2	Prisoner's dilemma like games	53
4.3	A simulation example	54
4.4	Simulating four different games	55
4.4.1	Round robin tournament	56
4.4.2	A population tournament	57
4.5	Noisy environment	58
4.6	Conclusions	58
5	Characteristic Distributions in Iterated Games	61
5.1	Introduction	61
5.2	Characteristic Distributions - Definitions	62
5.3	Characteristic Distributions - an Example	65
5.3.1	The two-layered approach to games	66
5.3.2	An Example	66
5.4	Optimality in games	68
5.4.1	Optimal strategies in a game	68
5.4.2	Optimal games for a strategy	68

5.4.3	“No Free Lunch Theorem” for strategies	69
5.5	Conclusions and Future Work	69
6	Conclusions and Future Work	71
6.1	Conclusions	71
6.2	Future Work	73
A	Simulation Tool for Strategies	75
A.1	Introduction	75
A.2	The Strategy Language	75
A.2.1	The Syntax of a Strategy	75
A.2.2	Terminals	76
A.2.3	Non-terminals	76
A.2.4	BNF grammar for the Strategies	78
A.2.5	The strategy file	79
A.3	Running the SITS system	80
A.3.1	How to configure SITS	80
A.3.2	How to start SITS	82
A.3.3	How to run series of simulations	83
B	Implementation of the strategies in SITS	85
C	Proofs	87
Lemma 1	87
Theorem 2	87
Theorem 3	89
Theorem 4	89
	Bibliography	91
	Indexes	97
Author Index	97
Subject Index	99

List of Figures

1.1	The society level of multi-agent systems	2
1.2	Game Theory at the society level	4
1.3	An example of a decision tree in an extensive game	5
1.4	A demographic game	12
2.1	A resource allocation problem	29
2.2	A comparison between PD and CG	34
3.1	A traffic intersection situation.	40
3.2	Proportions of outcomes for different strategies	43
3.3	Outcome for the strategies in PD and CG	46
3.4	Population distribution without noise	47
3.5	Four successful strategies in a noisy PD	49
3.6	Five successful strategies in a noisy CG	49
4.1	Characteristic distributions or “fingerprints”	54
5.1	The two-layered approach with characteristic distributions	62

List of Tables

1.1	Rock, Paper, Scissors payoff matrix	6
1.2	A coalition game	7
1.3	Examples of symmetries in games	8
1.4	Years in prison in prisoners dilemma	10
2.1	Prisoner's dilemma and chicken game on an ordinal scale	25
2.2	A general hawk-and-dove payoff matrix	26
2.3	A hawk-and-dove payoff matrix	27
2.4	Prisoner's dilemma and chicken game payoff matrices	30
2.5	Results from the parametrised hawk-and-dove game	33
3.1	Resource allocation as a time delay problem.	42
3.2	Descriptions of the different strategies.	44
3.3	Cost matrices for resource allocations	45
4.1	A basic 2x2 matrix	56
4.2	Payoff matrices for PD, CG, COG and CD	56
5.1	Some characteristic distributions	66
5.2	Yet another payoff matrix for the prisoner's dilemma	67
A.1	A BNF grammar for the strategy language	78
A.1	A BNF grammar for the strategy language	79
A.2	The Payoff matrix for a symmetric, two-strategy, two-choice game	81

Chapter 1

Introduction

Game Theory and Agents. The title of this work is chosen based on my point of view that the two areas of multi-agent systems (MAS) and game theory have an interesting intersection. We may see all behavioral patterns in interactions between agents in a MAS as a result of players acting according to strategies in a game. When taking that position, the problem is to develop environments that encourage “good” agent behavior (i.e. to define the game in a way that the agents choose the kind of strategies that we prefer), rather than hard-coding behavior protocols into the agents themselves.

This introduction will start by positioning game theory in the context of the society level in MAS’s. Then an overview of the field of game theory is given, followed by some comments on the intersection between the fields, i.e., in what cases we may use game theory in a purposeful way in MAS’s. At the end of the chapter there is an outline for the rest of the thesis.

1.1 The society level of MAS

The *society level* view (as proposed by e.g., Gustavsson [38] and Jennings [44]) of MAS’s is a way to partition different aspects of the system into three components. The hypothesis is that each MAS consists of *Agents*, *Coordination Mechanisms*, and a *Context* as illustrated in figure 1.1 below.

Since the area of agent research is one of the younger ones in the computer science discipline and still evolving, there are probably as many opinions of what an agent is, as there are researchers, something that is stated in e.g. [79]. However, there are also some characteristics of agents that can be agreed upon. Here are some examples of definitions of an agent:

- An *agent* is an encapsulated computer system that is situated in some envi-

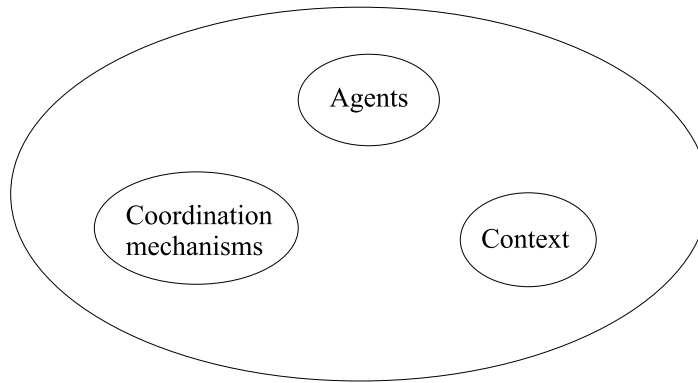


Figure 1.1: The society level of multi-agent systems. A MAS consists of agents, coordination mechanisms and a context.

ronment, and that is capable of flexible autonomous action in that environment in order to meet its design objectives. (Wooldridge [95])

- (Agents are...) components that communicate with their peers by exchanging messages in an expressive *agent communication language*. (Genesereth and Ketchpel [35])
- An agent will set out to do something, and do it; therefore it has the competences for intending to act, for action in an environment and for monitoring and achieving its goals. (Watt [90])
- I consider an agent to be an entity that is capable of acting in its environment to satisfy its desires. (Durfee [25])
- From the computing perspective, agents are autonomous, asynchronous, communicative distributed and possible mobile processes. From the AI perspective, they are communicative, intelligent, rational and possibly intentional entities. (Pitt and Mamdani [69])
- An *agent* is a meta-strategy that chooses strategies for playing games. (Johansson, chapter 5 [45])

As can be seen, the definitions of agents often refer to two or even all of the three components of the society level. We may, as in the case of the definitions by Wooldridge, Watt, Durfee and Johansson, ascertain that they all refer to agents as

something that perform actions *in an environment*.¹ Others, e.g. Genesereth and Ketchpel, and Pitt and Mamdani, defines agents as entities with *communicative skills*. Some may include agent properties in the definitions, such as desires and intentions to achieve a certain goal, as in the definitions of Watt, Durfee, and Pitt and Mamdani.

The society level distinguish these three perspectives in order to be able to discuss and compare the use of different technologies and methods when solving problems in a structured engineering way in the different components.

1.1.1 The coordination mechanism component

The coordination mechanism (or *high-level interactions*) is the component that describe under which conditions *interaction* between agents is possible. This may be mechanisms for auctions [82, 97], or negotiations [53, 54]. It may also be mechanisms used to coordinate cooperative tasks, as in e.g., contract nets [86] and multi-agent planning and organizational structures [92].

In the sense of *game theory*, we consider the *structure* of a game, to be the coordination mechanism from the society level point of view. By the structure of a game, we mean everything that describe the outward form of the game, such as the number of participating strategies, the number of choices that they have, and whether the game is strategic or extensive, i.e. the shape of the payoff function.

1.1.2 The context component

At the context level, the organizational aspects of the system is modeled. These include (but are not limited to): different *roles* that agents may take in a MAS (such as, sentinel agents [39]), social conventions, such as laws, restrictions [24], and coalition norms [45] that agents agree upon and penalties and rewards connected to these conventions. Also different properties of the external world belong to the context.

In game theoretic terms, we speak of context as being the (external) factors that may affect an agent's choice of strategy, e.g., the payoff for a certain combination of actions made by the participants, the expected length (and distribution of lengths) of an iterated game, and the type and level of noise.

By changing the context, we may change the opinion about what kind of behaviors that are normative in a MAS and the same is, of course, true in the case of game theory, – if we change the payoff matrix, it will likely have an effect on which strategies that will perform best in a game. Since rational agents always will

¹In the case of Johansson, the game is constructed in such a way that it corresponds to the environment.

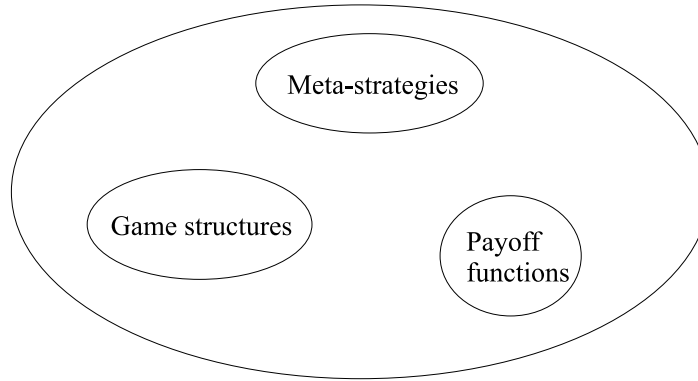


Figure 1.2: Game Theory at the society level. The agents are meta-strategies, choosing strategies for playing certain games (coordinating) given some payoffs (i.e. a certain context).

choose a strategy that gives the highest expected utility, we may, without knowing how a certain agent is implemented, guide its behavior towards following a norm by adjusting the payoff levels, i.e. changing the context.²

1.1.3 The agent component

Much can be said about different types of agents, e.g. concerning their reactivity and pro-activeness [18], rationality and emotional capabilities [19], and communicative skills [43]. However, the focus of this thesis is game theory and its applicability in multi-agent systems and we will therefore assume that the agents of the system are rational and thus utility-maximizing. They will therefore choose strategies in a way that favors the strategy that pays the best according to the context in which it is situated. The fact that the context is essential to the ability to choose appropriate actions has been pointed out e.g. by Wolpert and Macready in their “*No free lunch*” article. It states that all successful search methods are context-dependent, i.e. they take advantage of the structure in the search space in order to find the solution faster [94]. For the same reasons, agents must take their environment into consideration if they want to perform successful actions. That is, the question of whether an action is *successful* or not is fully dependent on the context in which it is performed.

²It would not be out of place make the reservation that we in practical applications often do not have control over the situation, i.e. we may not have all facts needed to determine the payoffs.

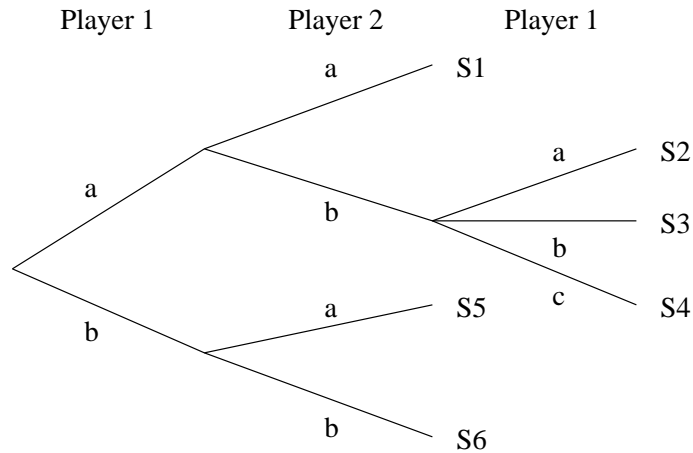


Figure 1.3: A decision tree for an extensive game with two players. First *Player 1* has two choices, followed by the choice of *Player 2*. If they played *a* and *b*, *Player 1* will have three choices, else one of the states *S1*, *S5* or *S6* is reached.

1.2 Game Theory

Game theory is a mathematical tool for analyzing decisions. As such, the type of situation that we try to model, highly influence what features are needed in such a tool. In the next subsections (1.2.1 to 1.2.7) we will describe some aspects of game theoretic modeling.

1.2.1 Temporal Aspects of Games

We distinguish between two different types of games based on the time points when actions from the players are performed.

In *extensive games*, the players act in turns. It is thus easy to describe the playing of a game in the form of a decision tree in which the nodes correspond to the states and the paths to the nodes the series of actions from the players (see figure 1.3).

Typical examples of extensive games are different kinds of board games such as Chess, Go, and Othello and there are several algorithms dealing with the search of the states with the highest utilities, e.g. the well-known minimax algorithm (with variations) [75, 79].

Another example is the centipede game, in which each player, when she is in turn, can choose between to

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

Table 1.1: The payoffs for the players in Rock, Paper, Scissors. The entries show the payoffs for $(player_1, player_2)$.

1. Collect the kitty. This leaves the opponent(s) without any payoff and the one who collected gets everything.
2. Do nothing. The turn is simply given to the next player. This action will increase the payoff, so if it will be our players turn again (without any other player having collected the kitty), the payoff of a collect action will be higher than in the previous round.

There is a problematic issue of backward induction that becomes especially clear in the finite versions of this game. If in a two-player game, one of them knows that the game will end in the next move, the most rational choice is to collect. If the other agent knows about the first ones intentions to collect, the most rational thing to do is to collect the round just before the first agent and so on³.

Strategic games are games in which the players, in contrast to the extensive games, make simultaneous actions. A player in this type of game will not know what action its opponents will make.⁴ This type of games are not only of interest when we have a true simultaneous situation, such as in the game *rock, paper, scissors* illustrated in table 1.1.

The games may also serve as an approximate models of situations where the players have an opportunity to act independent of when their opponents act, but the effect of the action is not noticed until the opponents have made their next moves. An example of this latter situation is the case where agents form a coalition. They may then choose actions that are permitted, given the norms of the coalition, or they may choose forbidden actions, which may cause an exclusion of them from the coalition. Consider the following example:

³A discussion of this problem, solutions and critique can be found in e.g. *Grappling with the Centipede* [70].

⁴In a game involving a player p we will refer to the rest of the players as the *opponents* of p regardless of whether they cooperate or not.

	I	N	F
I	$1 + 2C$	$1 + \frac{1}{2}C$	1
N	$2 + \frac{1}{2}C$	2	$2 - C$
F	10	9	0

Table 1.2: A coalition game where C is the value of being in the coalition and entry (i, j) is the payoff of the player choosing action i when its opponents choose j .

Example 1

An agent A is a member of a coalition C in which each agent have three different choices:

1. To improve C (action I). This strengthen the coalition, but gives a rather slightly negative short-time payoff.
2. To do nothing (action N). This action will effect neither its standing in the coalition, nor its payoff
3. To perform a forbidden action (action F) that will harm the coalition. This will result in a short-time high payoff, but also an exclusion of the coalition

We have a situation as follows: The payoff for an agent of the actions is in order $\text{payoff}_F > \text{payoff}_N > \text{payoff}_I$. At the same time, the payoff for the coalition is in the reverse order. The agents will then have to evaluate the utilities of actively working for, being part of, or harm the coalition. An example of such a payoff matrix for two agents is given in table 1.2.

In the rest of the thesis we will mainly discuss symmetric games, but the generalization to asymmetric games is, in most cases, straight forward.

1.2.2 Symmetry in Games

There are several aspects of symmetry in games.

- The games can have a symmetry in their *structure*. This means that the number of choices of actions is the same for all players.
- The games may also be symmetric in their *payoff*, i.e. the players will receive the same payoff for the same actions.

Description of game	structure	payoff
A company and its customers	asymmetric	asymmetric
Battle of Sexes	symmetric	asymmetric
Resource allocation with asymmetric restrictions	asymmetric	symmetric
Prisoner's dilemma	symmetric	symmetric

Table 1.3: Examples of games that are symmetric from the structural and payoff points of view.

Note that we may have games that are symmetric in one type, but not in the other as in the case of *Battle of Sexes*⁵ and *Resource Allocation with asymmetric restrictions*.⁶ In *A company and its customers*⁷ and *Prisoner's Dilemma*⁸ the game is asymmetric and symmetric from both agent's points of view respectively (see table 1.2.2). We will generally refer to *symmetric games* as games being symmetric from both the structural and payoff points of view [60].

1.2.3 Information in Games

The players in a game may or may not have complete information about the pay-offs or the state of the game. First, we will give two examples of games with incomplete information, then some example of games with complete information.

An example, discussed by e.g. Raiffa [71] is a situation where a player is to choose between labeling an urn or not. If she chooses not to try, it will neither cost her anything, nor will she have the opportunity to win anything. The urns look exactly the same, but contains balls of two different colors. If the player manage to guess the correct label, she will win some money and if she make the wrong guess, she will loose some. The player may also choose to improve its

⁵Example due to Luce and Raiffa [60]. A married couple decide to go out one night. One prefers a concert, while the other one prefer to catch a movie. If they cannot decide, they will not go at all, while none of them would prefer staying home compared to the first choice of their better half.

⁶Two customers with identical preferences, but different amounts of resources, limiting the possibility for one of them to buy what she wants.

⁷There will be an business deal only if a seller and a buyer agree on the deal; if only one of them agree, it will have tried to put more effort in trying to close the deal, than the ones who reject it.

⁸see e.g. 2.4 on page 30

chances by paying some money to get to pick balls from the urn she is about to label. The amount of information is crucial to what action will be the optimal choice of the player. If she knows that the probability of an urn to be of type 1, it will improve the chances to label the urn correctly, as will the knowledge of the distribution of the colors of the balls in the different types of urns.

Another example of a game with incomplete information is the game of poker. A poker player will not in advance know if the cards she changes will improve the hand or not. It is simply not possible for the player to determine in what state the game is in before it is over. One possible solution is to keep a set of possible states and use randomized strategies to act optimal [50, 49]. For an extensive survey of repeated games with incomplete information, see e.g. the work by Aumann and Maschler [2].

Chess and Othello are good examples of games with complete information. Both players know the state of the game, i.e. where the pieces are placed on the board. Also the values of different states are known⁹.

1.2.4 Repeated vs Iterated Games

We may distinguish two kinds of repeated games. In an *evolutionary game*, the players do not know who they meet in a series of games. We may see it as if they have an equal probability of meeting every other player in the next round and that they have to choose a strategy based on that. An example of such a situation is the boarding of a flight at an airport¹⁰. Everyone want to board as soon as possible and everyone also know (or is supposed to know) that if the order of boarding proposed by the boarding counter is followed, it will take less time to get the people on the plane, than if they board in a random order. Each traveler then have to decide whether she will wait until her seat row is to board or if she is going to board before it is her turn. The *opponents* in this case are people that she will probably not meet again and she may therefore ignore the possibility that someone responds to her action with the same behavior next flight.

In *iterated games*, we have players that recognize its opponents and remember their previous actions. It is not necessary that a player meets the same opponent in consecutive rounds. Instead it can save the history of the game, play other opponents, and restore the history when the old opponent again faces the player. Or it may play several opponents in parallel. One example of this is an artificial agent being part of several coalitions at the same time. It may have different

⁹At least, the value of the end position of the game is known, even if the knowledge of *how* to get there may be unknown for a player; however this is a limit of the player as such, not an inherent property of the game.

¹⁰Example due to Hofstadter [41]

Prisoner 1:	Prisoner 2:	
	Not Confess	Confess
Not Confess	1 year each	10 years for 1, 3 months for 2
Confess	3 months for 1, 10 years for 2	8 years each

Table 1.4: Years in prison in prisoners dilemma

ways of behaving in different contexts, i.e. when interacting within the different coalitions it is being a part of.

One of the most well-studied and interesting games in game theory is the Iterated Prisoners Dilemma (IPD). The prisoners dilemma is due to A.W. Tucker and is interpreted as follows by Luce and Raiffa ([60] page 95):

"Two suspects are taken into custody and separated. The district attorney is certain that they are guilty of a specific crime, but he does not have adequate evidence to convict them at a trial. He points out to each prisoner that each has two alternatives: to confess the crime the police are sure that they have done, or not to confess. If they both do not confess, then the district attorney states that he will book them on some very minor trumped-up charge such as petty larceny and illegal possession of a weapon and they will both receive minor punishment; if they both confess, they will both be prosecuted, but he will recommend less than the most severe sentence; but if one confesses and the other does not, then the confessor will receive lenient treatment for turning state's evidence whereas the latter will get "the book" slapped at him. In terms of years in a penitentiary, the strategic problem might be reduced to the figures in table 1.4."

In a series of articles and an influential book, Robert Axelrod describes two contests that he announced in the beginning of the 80's on what strategy would win an IPD, when the strategies were to meet each other in a round robin tournament. The only thing the contestants knew for sure was that the *random* strategy was added to the population of strategies [3, 4, 5, 6, 7, 8]. The results of the tournament showed that *tit-for-tat*, the simplest strategy of them all submitted by Anatol Rapoport, won both tournaments, and a debate started whether *tit-for-tat* was the optimal choice of strategy for prisoners dilemma or not.¹¹

¹¹ A description of *tit-for-tat* is given in table 3.2 at page 44.

Critique was put forth on stability by e.g. Boyd and Lorberbaum [17] and Farrell and Ware [28]; Bendor, Kramer and Stout [11] and Molander [66] have discussed noisy IPDs and Marinoff question the consistency of the theoretical prescription and the empirical results [62].

Kraines and Kraines proposed Pavlov¹² as an alternative solution as it manages to avoid getting out of the state of mutual defection [52]. Lately, Bendor and Swistak have written about the conditions under which a population of IPD strategies may converge [12].

An interesting feature of iterated games is if the players know *when* the game will end. The most rational action is then to try to maximize the payoff in the last round, even if it will harm the opponent, because the opponent will not have any chance to revenge the defection. However, if the player suspect that the opponent will defect in the last round as itself plans to do, it would pay better to defect one iteration earlier. This is what we call backward inductive reasoning and is discussed by Selten (*the Chain Store Paradox* [85]) and in the context of IPDs by Schuessler [83].

1.2.5 Noisy Games

When players interact, their conception of what their opponent did and their ability to perform the action that their strategies define might be distorted. This is what we call *noise* and we can recognize two kinds of such distortions, *the trembling hand* noise and *misinterpretations* [7, 84].

In the former case, when a player make the wrong action,¹³ all players in the game observe the actual performed action. This means that they all agree on what actions that have been performed.¹⁴

In the latter case, the player may not have chosen the “wrong” action, yet it will be interpreted as such by at least one of its opponents. This also means that the players will not keep the same history of the game since they have different opinions of what happened.¹⁵

A good example of the trembling hand noise is when the implementation of an action has failed (and occasionally will result in another action). An example of the misinterpretation kind of noise is when the communication between

¹²for a description, see *Simpleton* in table 3.2

¹³Consider a *wrong action* to be any action other than the one that the player is supposed to do according to its strategy.

¹⁴The expression *trembling hand* comes from the metaphor of the player having to choose between pressing buttons, but since her hand is trembling, she might (with some probability) by mistake press the wrong button.

¹⁵We leave the possibility of the own player to misinterpret its move, when misinterpreted by the others, out of account.

A_2	A_1	A_2
A_1	A	A_1
A_2	A_1	A_2

Figure 1.4: An example of a demographic game. The agent in the middle, A , have an environment where the probability of meeting one of the agents A_1, A_2 is inverted proportional to the distance to it; thus it is more probable that it will meet A_1 than A_2 .

the players for some reason is corrupt and from time to time will report wrong actions to the opponents. Both of these situations may have large impact on the outcome of a game, especially iterated games, in which vendettas originating from a misinterpretation or a trembling hand may be harmful to strategies such as the well-known *Tit-for-tat* [10, 11].

1.2.6 Situated Games

We know from e.g. biology and political science that a strategy that performs well in certain environments is less successful in others, e.g. telling jokes about blondes in the mens changing-room vs. telling them to a blonde female police in duty. Lately, there have been an growing interest in simulating games where the players are placed in some kind of demographic environment (see figure 1.4).

It is of less interest whether they have two, four or seventeen neighbors to play against, instead it provides an important tool for studying local interactions in a population of agents. The closeness between agents on the grids in the simulation may be of a physical kind, as well as social.¹⁶ Santa Fe Institute have been one of the driving forces in this research. Lindgren and Nordahl have discussed the issue of evolutionary models for spatially situated prisoner's dilemma games [56, 57] and Epstein and Axtell have written about the connections to the area of artificial life [26, 27]

¹⁶I.e. two agents may be considered to be close if they share the same preferences, are part of the same coalitions, or simply for other reasons have to interact.

1.2.7 Evolutionary Game Theory

Evolutionary game theory deals with the problem of finding good and stable equilibria in evolutionary games.¹⁷ There have been several titles released during the last years that treat this matter, e.g. by Weibull [91], Samuelson [80], and Hofbauer and Sigmund [40].

Through different mathematical tools such as replicator dynamics, analysis through Markov models¹⁸ and adaptive dynamics [68], the behavior of a population of rational agents may be traced, from its initial state, to an equilibrium (if it exists).

1.2.8 Meta-games

When the results of different strategies meeting each other is known, we may still be in doubt of what strategy to choose. This is especially the case in games where the most preferable solution is not stable, e.g. as in the case of the prisoners dilemma. The reason for this is that player one does not know whether the opponent will play what it thinks, or if the opponent has the same doubts as player one. The players can agree on sticking to a certain strategy, but if another strategy would score better, we may choose to defect the agreement and collect the higher payoff. Of course, agreeing on what strategy to choose in one game, is in itself another game, a *meta-game*. Abreu and Rubinstein have treated the balance between optimal payoff and minimal complexity in meta-games [1] and Sandholm and Crites have used reinforcement learning to improve the choice of strategies [81]. We will investigate meta-games further in chapter 5.

1.3 Applications of Game Theory

Game theory has been applied in several different areas such as biology, economics, political science, computer science, psychology and philosophy. This success is due to the fact that it models an important basis of interaction between entities in the modeled world. The fact that the *structure* of a game is context independent makes the game theoretic tool work whether it models interactions between societies, species, individuals, companies, bacteria or artificial agents.

Far from *all* problems are possible to solve with game theory. The belief that an interaction only will affect the agents that are participating directly in the

¹⁷A strategy or mix of strategies is evolutionary stable if an infinite homogeneous population adopting it cannot be invaded by mutants under the action of natural selection[64].

¹⁸A good beginners introduction to Markov models is given in *An Introduction to Natural Computation*, pp 215-227 [9].

interaction is in general a coarse simplification, done in order to decrease the size of the game. Also, the number of possible actions are often reduced, e.g. by simplifying a continuous domain to a finite discrete one. If these simplifications cannot be done, other methods, such as risk analysis may be considered, but that will not be discussed further here.

In biology, game theory may model such things as normal and paradox strategies in nature. For instance, the will of an individual to defend a territory is an evolutionary stable strategy, but so is its opposite, *not* to defend a territory, but to leave it if it gets invaded by another individual.¹⁹

In economics, game theory have been used to model economic decisions and this application is suitable for game theory in the sense that the utility of different actions often is measurable in monetary units and transferable. It may for instance be used in the example of the airplane landing rule described by Rosenschein and Zlotkin in their *Rules of Encounter* [76].

Example 2

Imagine a situation in which priority for airplane landing is given to planes with less reported fuel on board. Since the airlines may earn lot of money on getting the customers as fast as possible from airport to airport, both on less fuel consumption and a faster connection, they will be motivated to consistently under-report their fuel as they approach their destinations. However, if most airlines follow this strategy, they will outperform the honest ones and sooner or later, the situation will be that the control tower has to choose between two near-empty airplanes, one without fuel, and one with more left than it has reported.

Clearly, in this situation dishonesty pays. However to calm down the readers, we can tell that the airports regularly check the fuel levels when the airplanes have landed and those cheating will have to pay large fines. Other similar examples include tax paying and whether or not one should pay the TV license.

Also in the field of philosophy it is possible to use game theory to formalize situations of decision where ethic social aspects may be taken into account. To facilitate morality in societies, we have agreed upon written and unwritten laws and conventions. One example of such an unwritten law is the convention of queuing. There is no explicit law that tells people to line up e.g. when buying tickets, yet most people follow this social norm. If one person breaks the line, everyone else will have to wait for her, but if all people follows this example

¹⁹One of the rare appearances of this is a strategy adopted by a Mexican spider. When a spider is driven out of its hiding place, it may end up in a crevice already hosted by another spider, who immediately leaves in search for a new place and so on. This series of movements may continue for several seconds and sometimes the majority of spiders in the population will end up in new places [32].

they will all be worse off than if they queued [31]. Utility maximization is a debated matter in philosophy that goes all way back to Hobbes and even if it may be possible to find mixed strategies that are optimal in a population, it may not be acceptable from a moral point of view to swap between different actions by random [33].

Psychologists and sociologists use game theory e.g. in simulations that are made in hope of understanding social phenomena. A common setup is to, in a specific context, let agents interact in form of playing strategic games and then draw conclusions upon the results. Several volumes have been written on this subject, e.g. [24][88]. However it is a controversial question whether their results are applicable to other areas as well as within the social simulation society itself.

In political science, game theory has shown to be useful in designing policies and laws that are considered *fair* in their environment. As e.g. MacLean points out, the opinion of what is fair and what is not may vary from person to person (or country to country) and there are no solutions that will satisfy everyone [61]. One example of this is the problem of climate-warming gases such as CO₂ [58]. All countries agree on that the emission of CO₂ in the long run must be reduced in order to solve the problems with the ozone holes, but there are large differences in the possibilities of reducing their own emissions. The developing countries need help in this matter and there can be fundamental differences in the principles of fairness among the developed countries. How much help is fair that the latter give the former (in order for them to manage their emission quota) and what are reasonable levels of punishment for the countries breaking the emission deal?

To summarize, we may use game theory to (at some level) model interactions between agents. The information we get can be used in (at least) two ways:

- Agents may use the information in real time for the purpose of choosing the most appropriate behavioral pattern in a given context.
- Designers of MAS's may use information about different strategies in order to design the context in a way that it encourage a certain desirable type of strategies (i.e. behavior) in the system.

1.3.1 Game theory as a tool for modeling interactions between artificial agents

The use of game theoretic tools to model interactions between computerized agents has been a (yet quite sparsely used, but) growing method of formalizing the mechanisms of choice when mutual interests meet. In economics, politics,

cognitive sciences and biology, all dealing with non-formal agents such as people, companies, societies or animals and plants, game theory has been used to explain different phenomena. The criticism that computer scientists often put forth is that game theory as a tool is too simple, that it is hard to capture all parts of a decision in one payoff function, etc. However there are a number of reasons to believe that game theory is a fairly good tool for modeling, not only biological agents, but even better for artificial agents.

1. Computerized agents are deterministic (and therefore they can more easily be described) to a higher extent than the biological ones.
2. Often agents have some kind of utility function that is to be optimized, whereas people in general have a much more complex structure of their utilities.²⁰ Therefore it should be easier to study games in which programs play, instead of games where people's decisions are involved.
3. Computers are more fit to mechanically calculate their utilities than people are (in the case that people bother at all to calculate these numbers).

All since Axelrod's tournaments in the beginning of the 80's, many simulations have focused on 2x2-games²¹ in general and the two-person prisoner's dilemma in particular. This is a little bit unfortunate, since there are many *other* interesting games that have been put into the shade by the discussions about Tit-for-Tat's be or not to be. In chapter 2 we argue that the chicken game may be as interesting to study as the prisoner's dilemma and will to an higher extent reward a non-defecting behavior than the prisoner's dilemma.

As we see it, a game is *not* set once and for all; the payoff of different actions taken by the agents (or by its behavioral schemas – the strategies) is as complex as the environment of the agents. Several factors have an impact of the payoff of an outcome.²² As this was not enough, the action chosen by one agent will only be one dimension in the payoff matrix, the rest is decided by the actions of the other agents.

One example of a situation where the payoff matrix is dynamically determined by the environment is the following:

Example 3

Imagine a market of goods that have to be delivered immediately (e.g. an open electricity market) with several buyers as well as sellers. The sellers declare at

²⁰In fact I argue that they *must* have this utility function, explicitly or implicitly expressed in order to be able to make a decision of what to do [45].

²¹A two-player, two-choice symmetric game

²²An outcome is, as defined in chapter 5, a combination of actions (that leads to an entry in the payoff matrix).

each point in time, the amount of goods they have for sale. The market then distributes the buyers to the sellers in a way that level out the differences between the sellers of what they have left of their capacity.²³ What makes this situation interesting is the situation when there are more capacity to sell, than there are buyers. In this case, the seller may choose to report more capacity for sale, than it actually has left, in order to get a larger share of the market and to reduce its own overhead of unused capacity. Of course, since it does not know how much the other sellers will offer, it may end up getting *too* many customers and a demand for resources that exceeds its actual capacity, i.e. it will get overloaded. Such situations are of course not good, neither for the overloaded selling agent, nor for the system as a whole.

One way of preventing this situation is to give the over-bidders a punishment (whenever they are detected). Unfortunately the probability of detection is dependent upon several things, such as the bids from the others (if all sellers over-bids, it is the same status quo as if all were honest) and the actual demand of the market. However, if we, as designers of a system may find a punishment function that, in every situation punishes the over-bidders when they are detected in such a way that it cost them more, than they would gain by risking getting detected, then we would have solved the problem.

We must not be stuck in the belief that game theory is all about selecting strategies. From the multi-agent perspective, the design of contexts (or *games*) is just as important.

1.4 On the Scientific Method

In the beginning of my thesis work, the goal was to find an answer to the question what types of games were important when modeling agent interactions. Soon it showed that it all depends on the situation that the agent finds itself in and that the situation may depend on such things as the amount of available resource.

To be able to try out all kinds of strategies, the search went for a suiting simulation tool and the first simulations concerning parameterized games were run in the spring of 1997. Several simulations of each parameter setup were run and linear regression was used to calculate its performance correlation with the varied parameter, the cost of mutual defection.

Soon it was realized that there were interesting aspects of parameterized games, such as the characteristic distributions (see chapter 5) that were hard to implement in the simulator we used, and the decision to implement a tool,

²³This distribution is done implicitly by the market forces.

(SITS, described in appendix A). that better suited the simulations was made. It was for efficiency reasons implemented in C, but with a good modular structure making it fairly easy to extend as our demands increased. What the results of the simulations run on that system is concerned, an *anytime*-point of view have been taken. Let us explain this a little closer. The result of an iterated game is, as we see it, *not* the score given to each player, but the characteristic distribution of the players, i.e. the distribution of combinations of choices made by my player and its opponents. These distributions between two strategies may be calculated analytically, determined by simulations (to an arbitrary granularity) or looked up in a table.²⁴ We argue that the figures we present are of less interest than the knowledge of that we always may improve them when the situation demands it.²⁵

What the references are concerned, we have tried to trace results to their origin when possible and we find the original text readable. L^AT_EX and its B^IB^TE_X utility has been of great help in the writing procedure and in keeping track of the references [37, 55].

1.5 Outline of the Thesis

The following subsections present (when applicable) publication notes for the chapters of this thesis together with abstracts of them. Some minor modifications have been done of the published articles. Besides obvious typos, inter-paper references have been completed with section references and all papers have common lists of figures and tables, indexes and bibliography.

1.5.1 An Iterated Hawk-and-Dove Game

Chapter 2 was written together with Bengt Carlsson and presented at the *Distributed Artificial Intelligence '97* workshop in Perth, Australia, December 1, 1997. In 1998 it was published in Springer Verlag's *Lecture Notes in Artificial Intelligence*-series [21].

A fundamental problem in multi-agent systems is conflict resolution. A conflict occurs in general when the agents have to deal with inconsistent goals, such as a demand for shared resources. In chapter 2 we investigate how a game theoretic approach may be a helpful and efficient tool in examining a class of conflicts in multi agent systems.

²⁴If assumed that data of previous meetings under the same conditions are available.

²⁵Of course, since the simulation tool is fast, we have had no problems in running enough simulations to get high quality figures for our papers. A typical outcome of two strategies is based on 5000 meetings, each of, say 200 iterations.

In the first part of this chapter, we look at the hawk-and-dove game both from an evolutionary and from an iterated point of view. An iterated hawk-and-dove game is not the same as an infinitely repeated evolutionary game because in an iterated game the agents are supposed to know what happened in the previous moves. In an evolutionary game, evolutionary stable strategies will be most successful but not necessarily be a unique solution. An iterated game can be modeled as a mixture of a prisoner's dilemma game and a chicken game. These kinds of games are generally supposed to have successful cooperating strategies.

The second part of the chapter discusses situations where a chicken game is a more appropriate model than a prisoner's dilemma game. The third part of chapter 2 describes our simulation of iterated prisoner's dilemma and iterated chicken games. We study a parameterized class of cooperative games, with these classical games as end cases, and we show that chicken games to a higher extent reward cooperative strategies than defecting strategies.

The main result of our simulation is that a chicken game is more cooperating than a prisoner's dilemma because of the values of the payoff matrix. None of the strategies in our simulation actually analyses its score and acts upon it, which gave us significant linear changes in score between the games when linear changes were made to the payoff matrix. All the top six strategies are nice and have small or moderate differences in scores between chicken game and prisoner's dilemma. The worst eleven strategies, with a lower score than random, either start with defect or, if they start with cooperation, are not nice. All of these strategies are doing significantly worse in the chicken game than in the prisoner's dilemma.

1.5.2 Generous and Greedy Strategies

Chapter 3, written together with Bengt Carlsson and Magnus Boman, was originally presented at the *Complex Systems '98* conference in Sydney, Australia, December 3, 1998 [22] and is to be found in the proceedings of the conference.

We introduce generous, even-matched, and greedy strategies as concepts for analyzing games. A two person prisoner's dilemma game is described by the four outcomes (C,D), (C,C), (D,C), and (D,D), where the outcome (X,Y) is the probability of that the opponent acts Y, when the own player acts X.

In a generous strategy the proportion of (C,D) is larger than that of (D,C), i.e. the probability of facing a defecting agent is larger than the probability of defecting. An even-matched strategy has the (C,D) proportion approximately equal to that of (D,C). A greedy strategy is an inverted generous strategy. The basis of the partition is that it is a zero-sum game given that the sum of the proportions of strategies (C,D) must equal that of (D,C).

In a population simulation, we compare the prisoner's dilemma (PD) game

with the chicken game (CG), given complete as well as partial knowledge of the rules for moves in the other strategies. In a traffic intersection example, we expected a co-operating generous strategy to be successful when the cost for collision was high in addition to the presence of uncertainty. The simulation indeed showed that a generous strategy was successful in the CG part, in which agents faced uncertainty about the outcome. If the resulting zero-sum game is changed from a PD game to a CG, or if the noise level is increased, it will favor generous strategies rather than an even-matched or greedy strategies.

1.5.3 Modeling Strategies as Generous and Greedy in Prisoner's Dilemma like Games

The third paper of this thesis was the second one written together with both Bengt Carlsson and Magnus Boman and is included in chapter 4. It was presented at the *Simulated Evolution and Learning '98* conference in Canberra, Australia, November 28, 1998 and is currently in press for Springer Verlag's *Lecture Notes in Artificial Intelligence*-series [47]. In this thesis, an extended version of the paper is presented.

Four different prisoner's dilemma and associated games were studied by running a round robin as well as a population tournament, using 15 different strategies. The results were analyzed in terms of definitions of generous, even-matched, and greedy strategies. In the round robin, prisoner's dilemma favored greedy strategies. Chicken game and coordinate game were favoring generous strategies, and compromise dilemma the even-matched strategy Anti Tit-for-Tat.

These results were not surprising because all strategies used were fully dependent on the mutual encounters, not the actual payoff values of the game. A population tournament is a zero-sum game balancing generous and greedy strategies. When strategies disappear, the population will form a new balance between the remaining strategies. A winning strategy in a population tournament has to do well against itself because there will be numerous copies of that strategy. A winning strategy must also be good at resisting invasion from other competing strategies. These restrictions make it natural to look for winning strategies among originally generous or even-matched strategies. For three of the games, this was found true, with original generous strategies being most successful. The most diverging result was that compromise dilemma, despite its close relationship to prisoner's dilemma, had two greedy strategies almost entirely dominating the population tournament.

1.5.4 Characteristic Distributions in Iterated Games

The fifth chapter presents a formal description of the characteristic distributions and some implications of it. It is written in the form of a paper, although it has not yet been published.

In game theory, iterated strategic games are considered harder to analyze than repeated games (for which the theory of mixed strategies is a suitable tool). However, iterated games are in many cases more fit to describe the situation of computerized agents, since it take into account previous moves of the opponents, rather than just assigning each possible action a certain probability. We introduce the notion of characteristic distributions and discuss how it can be used to simplify and structure the analysis of strategies in order to provide a good basis for choosing strategies in games to come.

1.5.5 Conclusions

Chapter 6 concludes this licenciatur thesis with some discussion about the possibilities to use characteristic distributions in multi agent systems. Further more, an outline for the forthcoming work towards the PhD thesis is given.

1.5.6 The Appendices

The simulation tool, SITS, that was used in chapter 3 and 4 ([22, 47]) will be described in appendix A and the implementation of the strategies that we have used in the simulations are to be found in appendix B.

In appendix C, full proofs of the theorems that are presented in chapter 5 are given.

Chapter 2

An Iterated Hawk-and-Dove Game

2.1 Background

Conflict resolutions are typically resolved by an appropriate negotiation process. In a multi agent setting, there have been several proposals of such negotiation processes. A recent and promising approach are models of computational markets [82, 93]. In this case, and other proposals, the negotiation is modeled as an interaction protocol between the agents involved. An important issue in the agent theories is whether we have centralized control or not. In the latter situations we refer to the agents as autonomous.

We propose in this paper iterated games as models of decentralized conflict resolution. We thus propose and make a preliminary assessment of a game theoretical framework for establishing cooperative behavior between selfish agents.

The evolution of cooperative behavior among a group of self-interested agents has received a lot of attention among researchers in political science, economics and evolutionary biology. Prisoner's dilemma (PD) [60, 72] was originally formulated as a paradox where the obvious preferable solution for both prisoners, low punishment, was not chosen. The reason is that the first prisoner did not know what the second prisoner intended to do, so he had to guard himself. The paradox lies in the fact that both prisoners had to accept a high penalty in spite of that cooperation is a better solution for both of them. This paradox presumes that the prisoners were unable to take revenge on the squealer after the years in jail.

The iterated prisoner's dilemma is, on the other hand, generally viewed as the major game-theoretical paradigm for the evolution of cooperation based on reci-

procuity. This is related to the fact that the agents have background information about previous moves, an information that is missing in the single game case.

2.2 Disposition

There is a distinction between iterated games, like the iterated prisoner's dilemma (IPD), the iterated chicken game (ICG) etc., and evolutionary games like the hawk-and-dove game (HDG). In the iterated games there must be some background information while an evolutionary HDG is a repeated game without history. In the iterated games this knowledge is used in order to find the proper moves according to the strategies in question, while an evolutionary stable strategy (ESS) is the result of an evolutionary game. An iterated game can, contrary to the HDG, easily be simulated by using this knowledge. In section 2.3 and 2.4 these three games are compared and a distinction between evolutionary and iterated games are made.

A problem central to multi agent systems is resource allocation. We model a negotiation situation (section 2.5) between two agents wanting a resource as an IPD or an ICG and argue in favor of using an ICG as an equivalent or even better description. There are two arguments for doing this: firstly a resource allocation problem can sometimes be better described using a chicken game matrix and secondly ICG should be expected to be at least as cooperative as IPD.

Different IPD and ICG are compared in section 2.6 and 2.7 in a series of experiments. We find successive strategies in chicken game being more cooperating than those in prisoner's dilemma. Finally we conclude with a section on the implications of our results.

2.3 A Comparison Between Three Different Games

The three above mentioned games make the basis for this discussion of the applications of game theory in multi agent systems, the IPD, ICG and HDG. In common for all three games is that the players have two mutually excluding choices, to cooperate (C) or to defect from cooperation (D).

2.3.1 The Prisoner's Dilemma

The prisoner's dilemma (PD) is a well-studied game within the area of game theory and when iterated [6, 8]) it has, apart from being a model for cooperation in economical and biological systems, also been used in multi agent systems [59]. In the former disciplines, it has been used from a social science point of view

a			b		
	C_2	D_2		C_2	D_2
C_1	III, III	I, IV	C_1	III, III	II, IV
D_1	IV, I	II, II	D_1	IV, II	I, I

Table 2.1: Prisoner's dilemma (a) and chicken game (b) on an ordinal scale I-IV

to explain observed cooperative phenomena, while in multi agent systems it has been used to try to create systems with a predicted cooperative behavior. The payoff matrix for PD can be found in table 2.1a.

2.3.2 The Chicken Game

In the chicken game (CG) [78], the players payoff is lower when both of them play defect than what they would have received by cooperatively playing while the opponent defected. Its name originates from the front-to-front car race where the first one to swerve from the collision course is a "chicken". Obviously, if they both cooperate, they will both avoid the crash and none of them will either be a winner or risk their lives. If one of them steers away, they will be chicken, but will survive, while the opponent will get all the honor. If they crash, the cost for both of them will be higher than the cost of being a chicken (and hence their payoff is lower, see payoff matrix in table 2.1b). In the game matrices we use ordinal numbers $IV > III > II > I$ to represent the different outcomes¹.

2.3.3 The Hawk-and-Dove Game

The HDG is described as a struggle between "birds" for a certain resource [65] [64]. The birds can either have an aggressive hawk-behavior, or a non-fighting dove-behavior. When two doves meet, they will equally share the resource with a small cost or without any costs for sharing, but when meeting a hawk, the dove leaves all of the resource to the hawk without a fight. However, two hawks

¹It is possible to interchange rows, columns and players or any combination of these operations to obtain equal games. Prisoners dilemma and chicken game are two out of 78 possible games, representing different payoffs on an ordinal scale [73] p. 204. Ordinal scale means that only the orders of magnitude can serve as criteria in the taxonomy, in contrast to a cardinal scale which is based on the different values.

	C_2	D_2
C_1	$\frac{R-S}{2}, \frac{R-S}{2}$	$0, R$
D_1	$R, 0$	$\frac{R-F}{2}, \frac{R-F}{2}$

Table 2.2: HDG, R is the total resource, $\frac{R-F}{2}$ the average outcome of a hawk-hawk fight and $\frac{R-S}{2}$ the average outcome for two doves sharing a resource.

will fight for the resource until one of them gets hurt so badly that it leaves the fight. Table 2.2 shows the payoff matrix for a HDG. A dove (C) in the matrix is described as a cooperating agent and a hawk (D) as a defecting agent.

In the HDG, we can see that if $R > F > S > 0$, then we have a PD type of HDG, since $R > \frac{R-S}{2} > \frac{R-F}{2} > 0$ (which corresponds to PD's $IV > III > II > I$). As a matter of fact, the same is true for CG; when we have $R < F$, we get $R > \frac{R-S}{2} > 0 > \frac{R-F}{2}$ (which corresponds to CGs matrix). In a PD, a second condition is usually put on the values of the payoff matrix, namely that $2 \cdot III > I + IV$, that is: the resource, when shared in cooperation, must be greater than it is when shared by a cooperator and a defector. This condition was, as we see it, introduced for practical reasons more than for theoretical ones and we think that it may have done more harm than good to the area and its applications.

"The question of whether the collusion of alternating unilateral defections would occur and, if so, how frequently is doubtless interesting. For the present, however, we wish to avoid the complication of multiple 'cooperative solutions'."

Rapoport and Chammah 1965, [72] p. 35

Now if we remove this constraint and let $2 \cdot III \leq I + IV$ be another possibility, we have a true HDG situation with equality when $S = 0$. This means that we can describe every such PD as a HDG, just by transposing the PD payoff matrix to a HDG one. The case when $S = 0$, when the sharing of a resource between two cooperators/doves does not cost anything, has been the main setup of our simulations and the reason for that is that we find it a natural way of describing resource sharing in multi agent systems. The resource neither grows nor decreases by being shared by two agents compared to when a defecting agent or a hawk takes it from the cooperator/dove.

	C_2	D_2
C_1	$R/2, R/2$	$0, R$
D_1	$R, 0$	$(R - F)/2, (R - F)/2$

Table 2.3: A hawk-and-dove payoff matrix

2.4 Evolutionary and Iterated Games

Evolutionary games use the concept of Evolutionary Stable Strategies (ESS). For a strategy to be stable it requires that, if almost all members of the population adopt it, the fitness of these typical members are greater than that of any possible mutant. An ESS is a stable strategy in that if all the members of a population adopt it, then no mutant strategy could invade the population under the influence of natural selection [65]. Or in other words: an ESS is a strategy which does well against copies of itself. A successful strategy is one which dominates the population, therefore it will tend to meet copies of itself. Conversely, if it is not successful against copies of itself, it will not be able to dominate the population. In a hawk-and-dove game without any costs for two doves sharing the resource we have:

In this case it requires (with the notation $E(p, q)$ used to express the payoff to an individual adopting strategy p against an opponent adopting strategy q), for a strategy to be an ESS (Maynard Smith and Price in [65]) that either:

$$E(p, p) > E(q, p) \text{ or} \quad (2.1)$$

$$E(p, p) = E(q, p) \text{ and } E(p, q) > E(q, q) \quad (2.2)$$

This implies:

- C is not an ESS because $\frac{R}{2} < R$; a population of cooperating agents can be invaded by a defecting agent.
- D is an ESS if $\frac{1}{2}(R - F) > 0$ or $R > F$. This is the same as the solution to the single game prisoner's dilemma. The point is that an ESS is not restricted to one move, the game can be repeated an infinite number of times.
- A proportion of C and D is an ESS if $R < F$ because $\frac{1}{2}(R - F) < 0$ and $\frac{1}{2}R < F$ excludes both C and D from being a pure ESS. Instead there will

be a probability p for C and a probability $q = 1 - p$ for D. This is the chicken game state which easily can be determined by calculating its Nash equilibrium - the point where no actor unilaterally wishes to change his choice.

$$p = \frac{R - \frac{1}{2}R}{R + 0 - \frac{1}{2}(R - F) - \frac{1}{2}R} = \frac{R}{F} \quad (2.3)$$

and

$$q = 1 - \frac{R}{F} \quad (2.4)$$

In the evolutionary approach an agent must only know his own payoffs for different moves. No common knowledge, rationality, about the other agents is needed. This means that we don't have to explain how agents know that a Nash equilibrium will be reached. The problem is that this is not the same as finding a successful strategy in an iterated game where an agent must know something about the other's choice. A large amount of outcomes in both IPD and ICG can be the consequence of a rational, Nash, equilibrium. This is known as the *Folk Theorem*.²

To use a deterministic ESS will not be the best strategy if the other strategies can make a commitment based on knowledge instead. In a multi agent environment this knowledge about the other agent's choice makes it possible to simulate the game.

2.5 Examples of a Resource Allocation Problem

An every-day situation description of a game situation would be when two contestants get into conflict in a business matter. When the two strategies cooperate and defect the contestants can choose to share the result or one gives up when the contestant goes to court to settle the case. If the reality of the game is that both lose more by going to court than by giving up, we have a chicken game. If we instead have a lesser cost for going to court compared to giving up, under the assumption of unchanged other conditions, there will be a prisoner's dilemma situation. In a repeated chicken game with no background information we should intuitively expect at least the same or even stronger cooperative behavior to evolve in the chicken game compared to the prisoner's dilemma because of the larger costs of going to court.

²Lomborg ([59] p. 70-74) has an overview of this critique against Nash equilibrium and ESS.

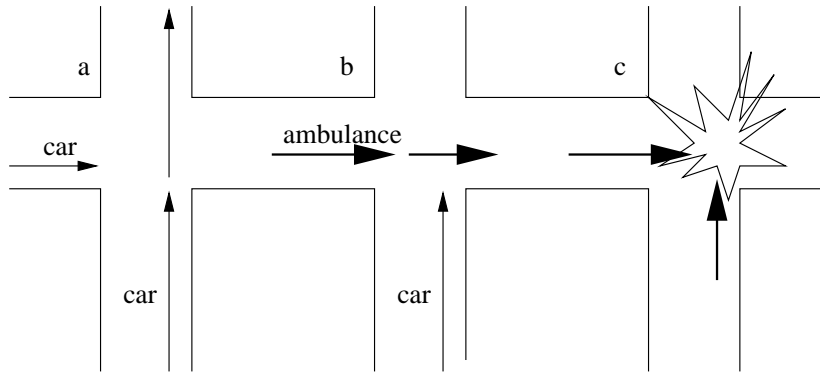


Figure 2.1: A resource allocation problem

Let us look at a traffic situation in an intersection using give right-of-way to traffic coming from the right (right-hand-rule). Drivers usually act in a cooperative mode and on average have to wait half of the time. No supervisor or central control is needed to have a functional system. Rescue vehicles, like the fire brigade or an ambulance, can however use an emergency alarm to get access to the lane. Let us suppose that if two ambulances both reach the intersection at the same time they will crash because they can't hear the siren from the other vehicle. If other cars begin to install sirens and behave as ambulances the whole traffic situation will collapse. The same thing happens if car drivers forget what is right and what is left. We call this random behavior a noisy one.

An analog to a traffic situation is how to model to get a whole resource where two agents normally share the resource half of the time each. We have the finitary predicament: real agents have only finite computational power and real agents have limited time.

If a very important or high priority agent wants the resource it will get it immediately and the other agent will get nothing. Two agents who want the resource at the same time, without having to wait for it, will cause a deadlock. If the cost for a deadlock is bigger than the cost for a cooperating agent meeting a defecting agent, then we have a chicken game, otherwise we have a prisoner's dilemma.

a			b		
	C_2	D_2		C_2	D_2
C_1	3,3	0,5	C_1	3,3	1,5
D_1	5,0	1,1	D_1	5,1	0,0

Table 2.4: Example payoff matrix prisoner's dilemma (a) and chicken game (b)

2.6 A Tournament Comparing Prisoner's Dilemma and Chicken Game

When Axelrod and Hamilton analyzed the iterated prisoner's dilemma they found out that a cooperating strategy, the Tit-for-Tat (TfT) strategy, did very well against more defecting strategies [6, 8]. This strategy has become an informal guiding principle for reciprocal altruism [87].

A TfT-agent begins with cooperation and then follows whatever the other agent is doing in a game lasting an unknown number of times. All agents are interested in maximizing individual utilities and are not pre-disposed to help each other. A defecting agent will always win or at least stay equal when meeting a TfT agent. In spite of that, a group of TfT agents will be stable against invasion of agents using other strategies because they are doing very well meeting their own strategy. No other strategy can do better against TfT than the strategy itself.³ TfT is a strategy that always repeats the example of the other strategy after the first cooperates. Depending on the surroundings this will be the best strategy, as in Axelrod's simulations, or a marginally acceptable or even a bad strategy.

As we see, the Hawk-and-dove game can be divided into two different game matrixes: The prisoner's dilemma-like game without the second condition of the prisoner's dilemma and the chicken game.

Axelrod found his famous Tit-for-Tat solution for the prisoner's dilemma when he arranged and evaluated a tournament. He used the payoff matrix in table 2.4a for each move of the prisoner's dilemma: The tournament was conducted in a round robin way so that each strategy was paired with each other strategy plus its

³Its true that TfT cannot be invaded by a defect, D, if there is sufficiently high concentration of TfT. But always cooperate, C, does as well as TfT in a population consisting only of itself and TfT, and hence can spread by genetic drift. This means that D can invade as soon as the frequency of C is high enough, since it has to fear less retaliation than against TfT alone. The game must be played an unknown number of iterations. The reason is that no strategy is supposed to be able to take advantage of knowing when the game ends and defect that iteration.

cont. Results from the simulations

Strategy	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0	I	N	S	Corr.	Diff.
sondeur dur	2.63	2.60	2.59	2.57	2.56	2.54	2.51	2.49	2.48	2.45	2.45	D		P	1.00	0.18
hesitante	2.38	2.39	2.40	2.41	2.41	2.43	2.45	2.46	2.47	2.48	2.49	C		X	-1.00	-0.11
majo mou	2.45	2.44	2.44	2.43	2.43	2.42	2.42	2.41	2.41	2.41	2.40	C	X		0.99	0.05
tf2t mou	2.45	2.44	2.44	2.43	2.43	2.41	2.40	2.40	2.41	2.38	2.38	C	X		0.96	0.06
pire en pire3	2.47	2.46	2.45	2.45	2.44	2.41	2.43	2.40	2.39	2.36	2.37	C	X		0.95	0.10
pavlov	2.35	2.36	2.35	2.35	2.35	2.36	2.35	2.35	2.35	2.36	2.35	C	X		-0.18	0.00
per gentille	2.22	2.24	2.26	2.28	2.30	2.32	2.34	2.37	2.39	2.41	2.43	C		X	-1.00	-0.21
sondeur4	2.49	2.46	2.41	2.37	2.33	2.29	2.25	2.21	2.18	2.14	2.09	C		P	1.00	0.40
c 4 sur 5	2.15	2.16	2.20	2.21	2.24	2.27	2.28	2.29	2.31	2.35	2.37	C		R	-0.99	-0.22
gentille	2.11	2.14	2.17	2.20	2.23	2.26	2.29	2.32	2.34	2.38	2.40	C	X	X	-1.00	-0.30
tf2t dur	2.33	2.31	2.29	2.28	2.27	2.25	2.23	2.22	2.20	2.18	2.16	C	X		1.00	0.17
sondeur2	2.25	2.25	2.25	2.26	2.25	2.25	2.25	2.25	2.25	2.26	2.25	D			0.11	0.01
slow TFT	2.35	2.33	2.31	2.29	2.27	2.25	2.23	2.21	2.18	2.17	2.14	C	X		1.00	0.20
mefiant	2.30	2.29	2.28	2.27	2.26	2.24	2.24	2.22	2.21	2.20	2.19	D			1.00	0.12
quatre c un t	2.09	2.12	2.14	2.17	2.20	2.22	2.25	2.27	2.30	2.32	2.35	C		X	-1.00	-0.26
rancuniere	2.42	2.38	2.33	2.29	2.25	2.20	2.16	2.11	2.07	2.03	1.99	C	X		1.00	0.43
ccctct	2.11	2.13	2.15	2.16	2.18	2.20	2.22	2.24	2.25	2.27	2.29	C		X	-1.00	-0.18
tft dur	2.35	2.32	2.28	2.25	2.22	2.18	2.14	2.11	2.08	2.04	2.01	C	X		1.00	0.34
lunatique	2.15	2.15	2.14	2.14	2.14	2.13	2.16	2.14	2.13	2.14	2.14	R		R	0.49	0.02
pire en pire	2.14	2.14	2.13	2.12	2.12	2.11	2.10	2.10	2.10	2.09	2.08	C		R	0.99	0.06
sondeur	2.25	2.23	2.19	2.17	2.14	2.10	2.06	2.01	2.01	1.96	1.91	D			1.00	0.34
per mechante	2.11	2.10	2.08	2.07	2.05	2.04	2.02	2.01	2.00	1.98	1.97	D		X	1.00	0.14
gradual killer	2.07	2.05	2.04	2.03	2.01	1.99	1.97	1.95	1.93	1.92	1.90	D		P	1.00	0.17
continued on next page																

2.6. A Tournament Comparing Prisoner's Dilemma and Chicken Game 33

cont. Results from the simulations

Strategy	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0	I	N	S	Corr.	Diff.
sondeur3	2.17	2.12	2.09	2.06	2.02	1.98	1.94	1.91	1.86	1.83	1.79	D		P	1.00	0.38
majo dur	2.09	2.06	2.05	2.02	2.00	1.97	1.95	1.92	1.91	1.89	1.86	D			1.00	0.23
calculateur	2.14	2.09	2.06	2.01	1.97	1.93	1.87	1.82	1.79	1.72	1.67	C			1.00	0.47
mieux en mie	1.91	1.90	1.88	1.88	1.87	1.87	1.86	1.85	1.85	1.84	1.83	D		P	0.98	0.08
pire en pire2	2.09	2.03	1.96	1.89	1.82	1.75	1.68	1.62	1.55	1.48	1.41	C			1.00	0.68
joss dur	1.93	1.90	1.86	1.82	1.78	1.72	1.66	1.64	1.59	1.56	1.50	C			1.00	0.43
mechante	1.86	1.78	1.70	1.63	1.55	1.47	1.39	1.31	1.23	1.16	1.08	D		X	1.00	0.78
average	2.30	2.29	2.27	2.26	2.26	2.24	2.22	2.21	2.20	2.18	2.17					

Table 2.5: Comparing prisoner's dilemma with chicken game.
Top columns: 1....0 - value of (D,D); I, N, S - characterization of the strategy; Corr. - Correlation coefficient (r); Diff. - The difference between basic prisoner's dilemma and chicken game.

Out of 36 different strategies *Gradual* won in a PD game. *Gradual* cooperates on the first move, then defects n times after n defections, and then calms down its opponent with 2 cooperation moves. In CG a strategy *Coop puis tc* won. This strategy cooperates until the other player defects and then alters between defection and cooperation the rest of the time. *TfT* was around 5th place for both games. Two other interesting strategies are *joss mou* (2nd place) and *joss dur* (35th place). Both start with cooperation and basically play *TfT*. *Joss mou* plays cooperation strategy one time out of ten instead of defect and *joss dur* plays defect one time out of ten instead of cooperate. This causes the large differences in scores between the strategies.

The top scoring games start with cooperation and react towards others i.e. they are not static. Both PD and CG have the same top strategies. A majority of the low score games are either starting with defect or have a static strategy.

Mechante (always defect) has the biggest difference in favor of PD and *gentille* (always cooperate) the biggest difference in favor of CG. The five games with the largest difference in favor of CG are all cooperative with a static counter. There

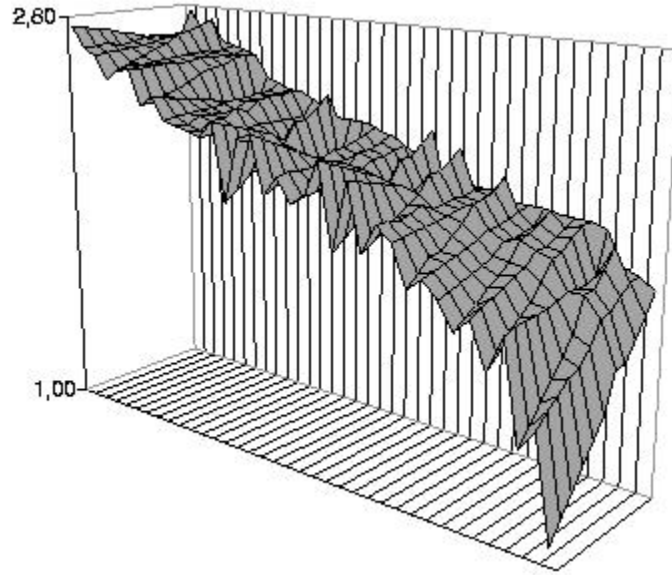


Figure 2.2: Comparing PD and CG. In the figure above the CG is in the foreground and the PD in the background, the best strategies are to the left and the worst to the right.

is no such connection for the strategies in favor of PD, instead there is a mixture of cooperate, defect and static strategies.

The linear correlation between the different kinds of games and the scores of each of the strategies were calculated. For all but six of the strategies there were a high confidence correlation value (r^2) exceeding 0.9. A minus sign before the r -value means that the strategy in question is more successful in the CG than in the PD (see table 2.5). For all these strategies there are significance levels exceeding the probability of 0.99. *TfT* is one of the remaining six strategies neither favored by PD nor by CG.

2.7 Discussion

We have shown the similarities between the hawk-and-dove game and the iterated prisoner's dilemma and chicken game. From a resource allocation point of view we argue that a parameterized game ranging from PD to CG is a suitable model for describing these kinds of problems.

This paper describes a simulation of iterated games according to Axelrod's ba-

sic matrix. Our simulation indicates that chicken game to a higher extent rewards cooperative strategies than the prisoner's dilemma because of the increased cost of mutual defections. These statements are confirmed by the following parts of the result:

1. All the top six strategies are nice and start with a cooperation. They have small or moderate differences in scores between chicken game and prisoner's dilemma. Tft is a successful strategy but not at all the best.
2. All the 11 strategies, with a lower score than *random (lunatique)*, either start with defect or, if they start with cooperation, are not nice. All of these strategies are doing significantly worse in the chicken game than in the prisoner's dilemma. This means that we have a game that benefits cooperators better than the prisoner's dilemma, namely the chicken game.
3. A few of the strategies got, despite of the overall decreasing average score, a better score in the chicken game than in the prisoner's dilemma (the ones with a negative difference in table 2.5). They all seem to have taken advantage of the increasing score for cooperation against defect. In order to do that, they must, on the average, play more C than D, when its opponent plays D. Here the mimicking strategies, like Tft, cannot be in this group, since they are not that forgiving. In fact, most strategies that demand some kind of revenge for an unprovoked defect, will be excluded, leaving only the static strategies.⁴ As can be seen in table 2.5, all static strategies which cooperate on the first move, and some of the partially static ones, do better in the chicken game than in the prisoner's dilemma. We interpret this result to be yet another indicator of the importance of being forgiving in a chicken game.

In a hawk-and-dove game we should expect defection to be an ESS in the prisoner's dilemma part of the game and a mixed ESS in the chicken game. In our simulation, defect (*mechante*) is among the strategies doing the worst and a mixture of cooperate and defect is not among the best strategies. To cooperate four times out of five (*c 4 sur 5*) corresponds in our simulation to a (D,D) score of 0.25 for an ESS and as we can see this is not at all the case because the ESS is a static and analytical concept. The reason for finding successful cooperating strategies is instead that the game is iterated, i.e. the relevant strategy "knows" the last move of the other strategy.

⁴In fact extremely nice non-static strategies (e.g. a *TfT*-based strategy that defects with a lower probability than it cooperates on an opponent's defection) would probably also do better in a Chicken game than in a prisoner's dilemma, but such strategies were not part of our simulations.

The only reason for *always defect* (*mechante*) and *always cooperate* (*gentille*) to vary a lot between prisoner's dilemma and chicken game is the changing payoff matrix. This is also the explanation why *TfT* and all the other strategies get other scores. If *TfT* meets another strategy it will mimic the others behavior independently of the values of the matrix. A defecting strategy will lose more in a chicken game because of the values of the matrix not because of a changing strategy. We are not at all surprised about finding very strong significance of a linear correlation in favor of PD or CG. This is exactly what these kind of strategies are expected to do.

There is no limit for the cost of mutual defection in a chicken game. Every strategy using defect will risk meeting another defect causing a high penalty. Strategies like always cooperate will be favored but they will still use the same interaction with other strategies.

2.8 Conclusions and Future Work

The hawk-and-dove game consists of two different game matrices:

- The prisoner's dilemma-like game does not fulfill the second condition of the prisoner's dilemma. When Rapoport and Chammah defined the second condition of the prisoner's dilemma, they wished to avoid the complication of multiple 'cooperative solutions'. In our opinion this was just a temporary restriction they made, not a definite one, as we argued earlier in this paper.
- The chicken game has a Nash equilibrium consisting of CD and DC. In an evolutionary approach this is called a mixed ESS. In an ICG this is not a sufficient solution because of the Folk Theorem.

A chicken game is more cooperating than a prisoner's dilemma because of the values of the payoff matrix. The payoff matrix in this first series of simulations is constant, a situation that is hardly the case in a real world application, where agents act in environments where they interact with other agents and human beings. This changes the context of the agent and may also affect its preferences. None of the strategies in our simulation actually analyses its score and acts upon it, which gave us significant linear changes in score between the games.

Another feature of this work is to clarify the role of general properties among strategies in a simulation. In this paper we look at three: initial move, how nice and static they are, but there are other aspects not covered here. Forgiveness may be an important factor in successful strategies in the chicken game and so may the ability to accept revenge.

It is impossible to simulate a hawk-and-dove game in an evolutionary context because of its randomized nature.⁵ An ESS is the result of a infinitely repeated game not the result of a simulated iterated game. Successful strategies in the iterated prisoner's dilemma and chicken game will be stable against invasion of other strategies because they are doing very well meeting their own strategy. No other strategy can do better against a successful strategy than the strategy itself.

An ESS is a strategy in that if all the members of a population adopt it, then no mutant strategy could invade the population. This means that after the simulation we can try to find such a successful ESS. *TfT* has been suggested to be an ESS because no other strategy can do better against *TfT* than the strategy itself. In practice it is hard to find such best strategies because of many equally good strategies and the possibility of genetic drift. What we found was that nice strategies starting with cooperate did very well against other strategies. Strategies with the lowest score either start with defect or, if they start with cooperate, are not nice. All of these strategies are doing significantly worse in a chicken game than in a prisoner's dilemma. This means that the chicken game part of the hawk-and-dove game suits cooperators better than the prisoner's dilemma part.

⁵It is possible to simulate the game through a process of selection, consisting of two crucial steps: mutation, a variation of the way agents act, and selection, the choice of the best strategies [56], but this is not within the scope of this paper.

Chapter 3

Generous and Greedy Strategies

3.1 Background

In the area of multi-agent systems (MAS), game theory [74] has proven useful, particularly as a tool for modeling the behavior of utility-based agents (see, e.g., [76]). In the quest for identifying and eventually inducing rational behavior in artificial agents, game theory has also been adopted as a normative theory for action. The main inspiration for this research has been the original axiomatic formulations of utility theory, starting with [89]. The difficulties involved in choosing a particular such axiomatization as a blueprint for agent simulations led MAS researchers to simplify the assumptions of game theory. Confusion about the usefulness in practice of game-theoretic approaches in some MAS papers has led to criticism (cf. [15, 23, 59]). That said, simulation methods in MAS have been successfully connected to utility theory and economics, and generally to reasoning under uncertainty, and MAS simulation has matured into an important subtopic (see, e.g., [16]).

3.2 Methodology

In section 3.3, we introduce a generous-and-greedy model for strategies. There are at least four different questions that should be addressed when trying to implement this model:

1. Which kinds of strategies are involved?
2. Which kinds of games are played?

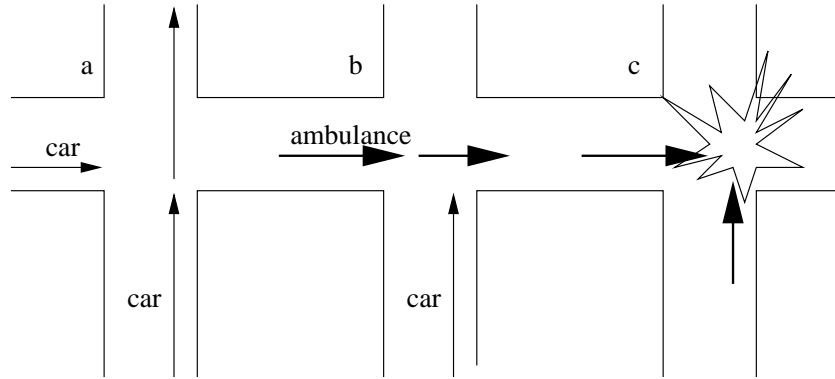


Figure 3.1: A traffic intersection situation.

3. What does a population of strategies look like?
4. What happens if the agents are uncertain about how to react against a strategy?

In sections 3.4-3.7 a traffic intersection example is described and simulated using both a population tournament and a noisy environment. We first look at questions 1 and 2, in section 3.5. Our main interest is to discuss dynamics, not to find the optimal solution for a certain kind of problem. We will look at 15 different strategies within two prisoner's dilemma-like games: the Iterated Prisoner's Dilemma (IPD) and the Iterated Chicken Game (ICG).

In section 3.6, question 3 is treated as a population tournament. We start with the same amount of agents for each strategy and let the different agents compete within a population tournament. Finally, in section 3.7, we look at question 4. Introducing noise into the strategies simulates the "*shaky hand principle*". This means that the strategy changes to the opposite strategy for a given percentage of moves. We conclude with a short section on the implications of our results.

3.3 A Generous-and-Greedy Model for Strategies

The PD is a well-studied game, used in MAS to create systems with a predicted cooperative behavior[59]. When Axelrod and Hamilton analyzed the IPD, they found that a co-operating strategy, called *Tit-for-Tat* (*TfT*), did very well against strategies with more defect ([6, 8]). This strategy has become an informal guiding principle for reciprocal altruism [87]. A *TfT* agent begins with cooperation

and then imitates its opponent, in a game of unknown length. Axelrod describes this as being nice and forgiving against a defecting strategy that uses threats and punishments. Binmore presents a critical review of *TfT*, and of Axelrod's simulation. He concludes that *TfT* is only one out of a very large number of equilibrium strategies and that it is not evolutionary stable ([13] p. 194-203). On the other hand, evolutionary pressures select equilibria for IPD in which the agents eventually tend to cooperate.

Instead of highlighting niceness or some other similar property, we will analyze strategy quality strictly through proportions of (C,C), (C,D), (D,C), and (D,D). The notation (C,D) means that the first agent is playing cooperate against a second defecting agent, etc. We will next define informally a partition of the strategies, as an alternative to Axelrod's incomplete interpretation, in terms of nice, resistant to provocation, and evil strategies.

A *generous strategy* cooperates more often than its opponents do when they meet. This means that the proportion of (C,D) is larger than that of (D,C), i.e. the probability of facing a defecting agent is larger than the probability of defecting.

An *even-matched strategy* has the (C,D) proportion approximately equal to that of (D,C).

A *greedy strategy* defects more often than its opponents do when they meet, making it an inverted generous strategy.

The basis of the partition is that it is a zero-sum game on the meta-level in that the sum of proportions of the strategies (C,D) must equal the sum of the strategies (D,C). In other words, if there is a generous strategy, then there must also be a greedy strategy. The classification of a strategy can change depending on the surrounding strategies. Let us assume we have the following four strategies:

Always Cooperate (AllC) has 100% cooperate ((C,C) + (C,D)) when meeting another strategy. *AllC* will never act as a greedy strategy.

Always Defect (AllD) has 100% defect ((D,C) + (D,D)) when meeting another strategy. *AllD* will never act as a generous strategy.

Tit-for-Tat (TfT) always repeats the move of the other contestant, making it a repeating strategy. *TfT* naturally entails that $(C,D) \approx (D,C)$.

Random plays cooperate and defect approximately half of the time each. The proportions of (C,D) and (D,C) will be determined by the surrounding strategies.

	Cooperate (C2)	Defect (D2)
Cooperate (C1)	$1.5T, 1.5T$	$2T, T$
Defect (D1)	$T, 2T$	$1.5T + qT, 1.5T + qT$

Table 3.1: Resource allocation as a time delay problem.

Random will be a greedy strategy in a surrounding of *AllC* and *Random*, and a generous strategy in a surrounding of *AllD* and *Random*. Both *TfT* and *Random* will behave as an even-matched strategy in the presence of only these two strategies as well as in a surrounding of all four strategies, with *AllC* and *AllD* participating in the same proportions. All strategies are even-matched when there is only a single strategy left. The described relation between strategies is independent of what kind of game is played, but the actual outcome of the game is a linear function of the payoff matrix.

3.4 A Traffic Intersection Example

Let us look at a traffic situation in an intersection using give right-of-way to traffic coming from the right (right-hand-rule). Drivers usually act in a cooperative mode and on average have to wait half of the time (figure 3.1a). No supervisor or central control is needed to have a functional system. Rescue vehicles, like the fire brigade or an ambulance, can however use an emergency alarm to get access to the lane (figure 3.1b). Let us suppose that if two ambulances both reach the intersection at the same time they will crash because they cannot hear the siren from the other vehicle (figure 3.1c). If other cars begin to install sirens and behave as ambulances the whole traffic situation will collapse. The same thing happens if car drivers forget what is right and what is left. We treat such behavior as noise.

Suppose it takes time T to cross the intersection. If an ambulance comes across a car, it will immediately get access to the lane. Two cars will on average need $1.5T$ to cross the intersection (we assume that there are no other time consuming delays). Two ambulances will get $1.5T + qT$, meaning that their disagreement will cause some extra costs.

Two similar games provide the foundations for this discussion of the applications of game theory in MAS: IPD and ICG. We could also have chosen, with a similar example, other PD like games like coordination game or compromise dilemma (see chapter 4 [47]).

We will use this traffic intersection problem as an example of how to dis-

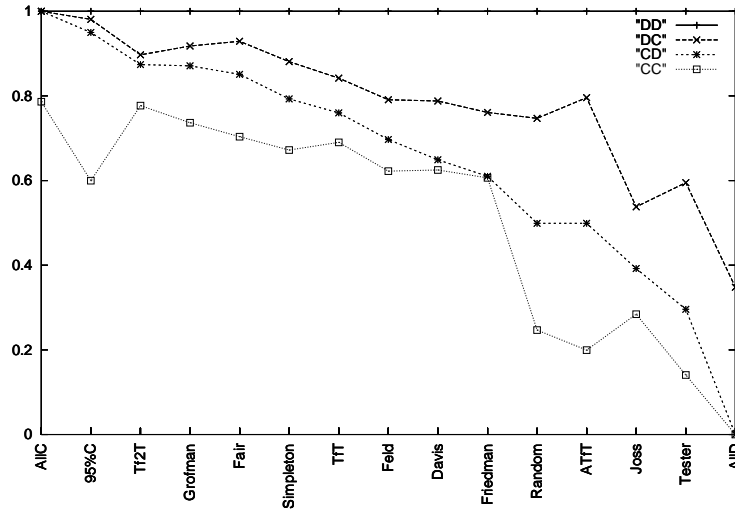


Figure 3.2: Accumulative proportions of (C,C) , (C,D) , (D,C) and (D,D) for different strategies. Note how the four actions partition the space of possible actions.

tribute (time) resources using a game theoretical model. Instead of ambulances we will talk about defecting agents that always want the resource immediately. The cars are cooperating agents that try to solve the resource allocation problem using the right-hand-rule.

3.5 Simulating the Traffic Intersection Example

For our simulation of the traffic intersection problem, we developed a simulation tool [46] in which 15 different strategies competed. Most of the strategies are described in ([3, 4], see also Figure 3.2). All strategies handle the moves of the other agent and not the payoff value, since the latter does not affect the strategy. In a round-robin tournament, each strategy was paired with each different strategy plus its own twin, as well as with the *Random* strategy. Each game in the tournament was played on average 100 times (randomly stopped) and repeated 5000 times (see Fig 3.2).

We interpret the proportions as a kind of fingerprint for the strategy in the given environment, independent of the actual value of the payoff matrix. For

Strategy	First move	Description
<i>AllC</i>	C	Cooperates all the time
<i>95%C</i>	C	Cooperates 95% of the time
<i>Tf2T</i>	C	<i>Tit-for-two-Tat</i> , Cooperates until its opponent defects twice, and then defects until its opponent starts to cooperate again
<i>Grofman</i>	C	Cooperates if (C,C) or (D,D) was played, otherwise it cooperates with a probability of 2/7
<i>Fair</i>	C	A strategy with three possible states – “satisfied” (C), “apologizing” (C), and “angry” (D). It starts in the satisfied state and cooperates until its opponent defects; then it switches to its angry state, and defects until its opponent cooperates, before returning to the satisfied state. If <i>Fair</i> accidentally defects, the apologizing state is entered and it stays cooperating until its opponent forgives the mistake and starts to cooperate again [56]
<i>Simpleton</i>	C	Like <i>Grofman</i> , it cooperates whenever the previous moves were the same, but it always defects when the moves differed (e.g. (C,D)).
<i>TfT</i>	C	<i>Tit-for-Tat</i> . Repeats the moves of the opponent
<i>Feld</i>	C	Basically a <i>Tit-for-Tat</i> , but with a linearly increasing (from 0 with 0.25% per iteration up to iteration 200) probability of playing D instead of C
<i>Davis</i>	C	Cooperates on the first 10 moves, and then, if there is a defection, it defects until the end of the game
<i>Friedman</i>	C	Cooperates as long as its opponent does so. Once the opponent defects, <i>Friedman</i> defects for the rest of the game
<i>ATfT</i>	D	<i>Anti-Tit-for-Tat</i> . Plays the complementary move of the opponent
<i>Joss</i>	C	A <i>TfT</i> -variant that cooperates with a probability of 90%, when opponent cooperated and defects when opponent defected
<i>Tester</i>	D	Alters D and C until its opponent defects, then it plays a C and then <i>TfT</i> the rest of the iterations
<i>AllD</i>	D	Defects all the time

Table 3.2: Descriptions of the different strategies.

a			b		
	C2	D2		C2	D2
C1	2 (a)	5 (b)	C1	1.5 (α)	2 (β)
D1	0 (c)	4 (d)	D1	1 (γ)	1.5+q (δ)

Table 3.3: A cost matrix for the Axelrod (a) and the resource allocation (b) matrices.

some of the strategies this is valid beyond doubt: As already noted, *AllC* and *AllD* have 100% cooperate ((C,C) + (C,D)) and 100% defect ((D,C) + (D,D)), respectively, while *TfT* entails that (C,D) ((D,C), for all payoff matrices.

AllC definitely belongs to a group of generous strategies and so do 95% *Cooperate* (95%C), *Tit-for-two-Tats* (Tf2T), *Grofman*, *Fair*, and *Simpleton*, in this specific environment.

The even-matched group of strategies includes *TfT*, *Random*, and *Anti-Tit-for-tat* (ATfT).

Within the group of greedy strategies, *Feld*, *Davis*, and *Friedman* belong to a smaller family of strategies doing more cooperation moves than *Random*, i.e. having significantly more than 50% (C,C) or (C,D). An analogous family consists of *Joss*, *Tester*, and *AllD*. These strategies cooperate less frequently than does *Random*.

What will happen to a particular strategy depends both on the surrounding strategies and on the characteristics of the strategy. For example, *AllC* will always be generous while 95%C will change to a greedy strategy when these two are the only strategies left. To see what these proportions mean to different payoff matrices, we recall our traffic intersection example and compare this to Axelrod's original matrix. Instead of using Axelrod's high score payoff matrix ((C,C)= 3, (C,D)= 0, (D,C)= 5, and (D,D)= 1), which we call *MaxAx*, we will use a low score matrix, *MinAx*, shown in Fig 3.3 a. If q in table 3.3 b is between 0 and 0.5 it is a PD game, and if $q > 0.5$ it is a CG.

The average payoff $E_{avg}(S)$ for a strategy S is a function of the payoff matrix and the distribution of the payoffs among the four outcomes (with the Greek letters referring to table 3.3 b):

$$E_{avg}(S) = p(C,C)\alpha + p(C,D)\beta + p(D,C)\gamma + p(D,D)\delta \quad (3.1)$$

We ran a simulation with the values for $1.5+q$ equal to: 1.6; 1.9 (PDs have dashed

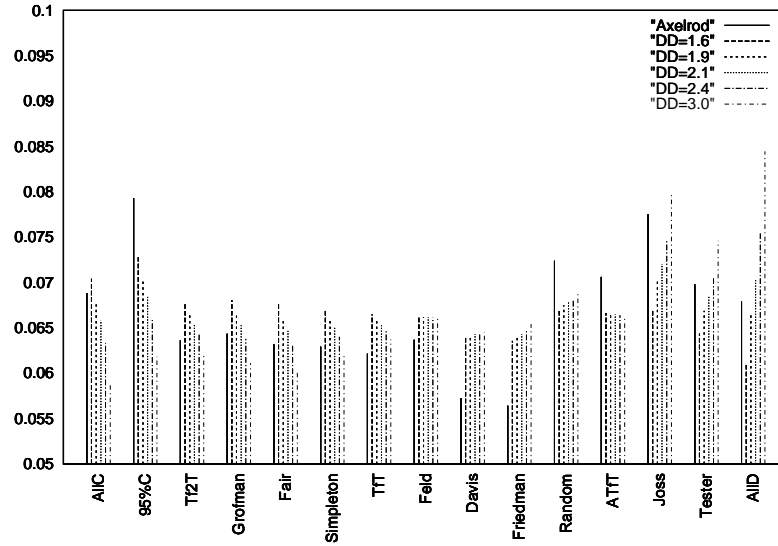


Figure 3.3: Outcome for the strategies in PD and CG. A lower score means a better result.

lines); 2.1; 2.4; 3.0 (CGs have dotted lines), and compared this to the MinAx matrix (the solid line), see Fig 3.3. A lower score means a better result for the strategy. For the MinAx case a correlated value to the MaxAx is obtained by adding $E_{avg}(MaxAx) - E_{avg}(MinAx)$. The result is normalized to 1 for the sum of all the strategies in each game. In this example, with 15 different strategies, each strategy gets the value 0.0667 on average.

None of the strategies in our simulation actually analyses its score and acts upon it. If we know the outcome of the competition between the strategies it is possible to calculate whatever payoff values are needed. This means that there is a linear correlation between the changes in scores between the games (see also chapter 2 [21]). Our choices of (D,D) are showing values near the borders 1.5 and 2.0 of the PD games and the border 2.0 of the CG. It is easy to extrapolate to another value, if desired. For all PD games (solid and dashed lines) there is a greedy strategy having a best score, but the result shows a large variation between different strategies. In the matrices of Axelrod and 1.9 PD, the strategies *Davis* and *Friedman* are doing best, while in 1.6 PD, *ALLD* is the winner. In CG, generous strategies are doing increasingly well with enhancements of the (D,D) value. This was expected, since there is an increase in the (D,D) value, and a linear payoff function was used.

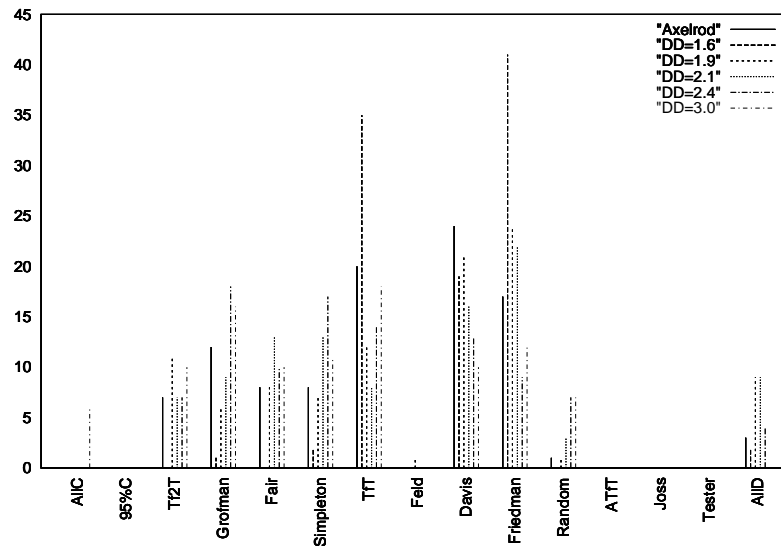


Figure 3.4: Population game without noise. Each bar shows the percentage of the total population for a strategy in a certain game.

3.6 A Population Tournament

Up until now nothing has been said about what happens if the number of agents within each strategy is allowed to vary. Maybe some vehicles after an unsuccessful trial want to change to a better strategy and ultimately find an optimal strategy for crossing the intersection. For our purposes it does not matter if we actually have ambulances and cars or if the vehicles behave like an ambulance in one intersection and as a car in another. A population tournament was held, letting each game continue until there was a single winning strategy left, or until the number of generations exceeded 10,000. For most of the games, one strategy won before reaching this limit (3150 generations were required on average). Each payoff matrix was used 100 times and the same (D,D) values were used as in the previous example. There were only four strategies not winning a single game (Fig 3.4). The most successful strategy was *Friedman*, which won the most games for three out of five different (D,D) values. Together with *Davis*, also a successful strategy, it belongs to the family of greedy strategies. For the PD part of the game *TfT* was successful. The generous strategies *Tf2T*, *Grofman*, *Fair*, and *Simpleton* form a rather successful family for the CG part. In Axelrod's matrix, the greedy strategies *Davis* and *Friedman*, together with *TfT*, are the winners. Notice that,

because of the zero-sum nature of the game, all winners must become even-matched at the end. The initial observation of different kinds of strategies shows us how the strategies reached this even-matched state, and eventually why they are successful.

3.7 Adding Noise

In the next simulation, we introduced noise on four levels: 0.01, 0.1, 1.0, and 10%. This means that the strategies changed to the opposite moves for this given percentage. The presence of uncertainty makes a huge difference as to which applications our results might have, and several writers (cf., e.g., [59]) have argued for the fact that noise as used here is an adequate representation of uncertainty. In Axelrod's simulation, *TfT* still won the tournament when 1% chance of misunderstanding was added [6]. In other simulations of noisy environments, *TfT* has instead performed poorly [10]. The uncertainty represented by the noise reduces the payoff of *TfT* when it plays itself in the IPD.

Instead of looking at all the different games we formed two different groups: PD, consisting of the Axelrod, 1.6D and 1.9D matrices, and CG consisting of 2.1D, 2.4D and 3.0D matrices. For each group we examined the five most successful strategies for different levels of noise. Fig 3.5 and 3.6 show these strategies for PD and CG when 0, 0.01, 0.1, 1.0, and 10% noise is introduced.

Among the four most successful strategies in PD there were three greedy and one even-matched strategy. In all, these strategies constituted between 85% (1% noise) and 60% (0.1%) of the population. *TfT* was doing well with 0.01% and 0.1% noise, *Davis* was most successful with 1% noise, and *AllD* with 10% noise.

Three out of five of the most successful strategies in CG were generous. The total line in Fig 3.6 shows that five strategies constitute between 50% (no noise) and nearly 100% (0.1% and 1% noise) of the population. *TfT*, the only even-matched strategy, was the first strategy to decline as shown in the diagram. At a noise level of 0.1% or more, *TfT* never won a single population competition. *Grofman* increased its population until 0.1% noise, but then rapidly disappeared as noise increased. The same pattern was shown by *Simpleton* that declined after 1% noise level. Only *Fair* continued to increase when more noise was added, making it a dominating strategy at 10% noise together with the greedy strategy *AllD*.

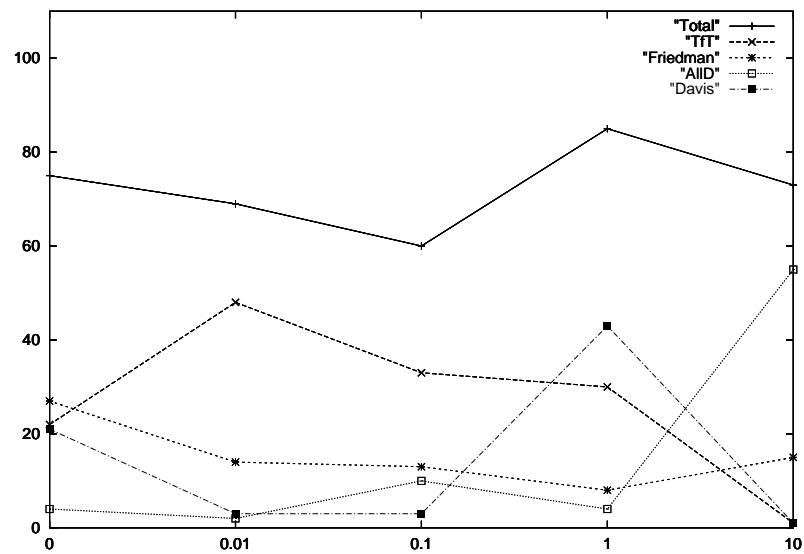


Figure 3.5: The four most successful strategies in PD games with increasing noise.

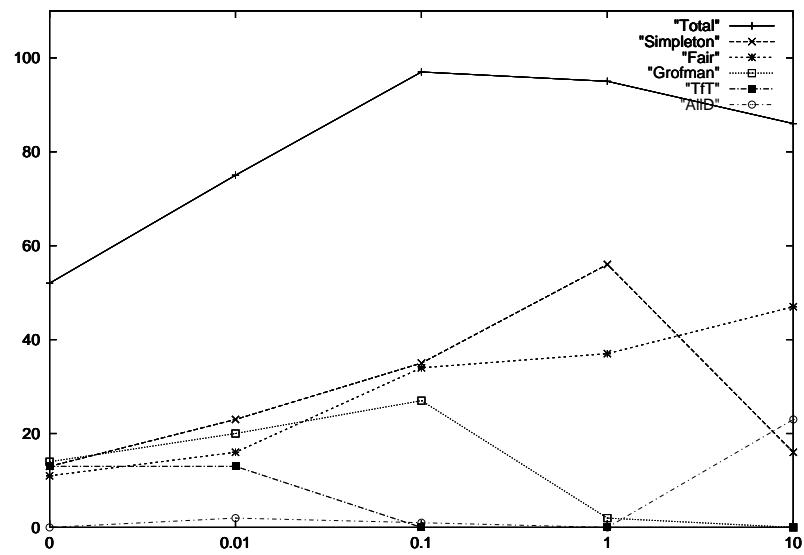


Figure 3.6: The five most successful strategies in CG games with increasing noise.

3.8 Conclusions

Having illustrated the concepts of generous, even-matched, and greedy strategies we now return to the four questions posed in section 3.2.

Which kinds of strategies are involved? Each strategy involved can be described using a "fingerprint" for each agent with a certain amount of (C,D) and (D,C) forming generous, even-matched, or greedy strategies. A new environment involves a new fingerprint for each agent.

Which kinds of games are played? The outcome of the game will depend on the payoff matrix involved. With the given interpretation of generous and greedy strategies it is natural to look at PD like games because they consist of co-operating and defecting behaviors. Different PD and CGs are the result of changes in the (D,D) value. For a certain set of strategies there is a linear correlation between the score of (D,D) and the score of each strategy.

What does a population of strategies look like? A successful strategy has to do well against itself so, if the cost of the (D,D) value is high, we should expect generous or even-matched strategies to be successful. In CGs, cooperation proved to be increasingly fruitful, following an increase in the (D,D) value from 2.1 over 2.4 to 3.0. For strategies competing in a round-robin tournament, greedy and even-matched strategies did well in PD games, with *Friedman*, *Davis*, and *TfT* outscoring the other strategies in our traffic intersection example.

What happens if the agents are uncertain about how to react against a strategy? We looked at an uncertain environment, free from the assumption of any existing perfect information between strategies, by introducing noise. Generous strategies were dominating the CG while greedy strategies were more successful in PD. In PD, *TfT* was successful with a low noise environment and *Davis* and *AllD* with a high noise environment. *Fair* was increasingly successful in CG when more noise was added.

We conclude that the generous strategies are more stable in an uncertain environment in CG. Especially *Fair* and *Simpleton* were doing well, indicating these strategies are likely to be suitable for a particularly unreliable and dynamic environment. The same conclusion about generous strategies in PD, for another set of strategies, has been drawn by Bendor ([10, 11]). In our PD simulations we found *TfT* being a successful strategy when a small amount of noise was added while greedy strategies did increasingly better when the noise increased. This indicates that generous strategies are more stable in the CG part of the matrix both with and without noise.

Given these results, and our chosen example, we recommend resource allocation agents to adapt a co-operating, generous strategy when the cost for a collision is high, or when different agents cannot be certain of the outcome of the game.

Chapter 4

Modelling Strategies as Generous and Greedy in Prisoners Dilemma like Games

4.1 Introduction

In multi agent systems the concept of game theory is widely in use. There has been a lot of research in distributed negotiation [34], market oriented programming [93], autonomous agents [76] and, evolutionary game theory [56] [59].

The evolution of cooperative behavior among self-interested agents has received attention among researchers in political science, economics and evolutionary biology. In these disciplines, it has been used from a social science point of view to explain observed cooperation, while in Multi Agent Systems (MAS) it may be used to try to create systems with a predicted cooperative behavior. In section 4.2 we look at prisoner's dilemma like games and the Tit-for-Tat (*TfT*) strategy.

In evolutionary game theory [64], the focus has been on evolutionary stable strategies (ESS). The agent exploits its knowledge about its own payoffs, but no background information or common knowledge is assumed. An evolutionary game repeats each move, or sequence of moves, without a memory. In many MAS, however, agents frequently use knowledge about other agents. We look at three different ways of describing ESSs and compare them to MAS.

Firstly we treat the ESS as a Nash equilibrium of different strategies. A Nash equilibrium describes a set of chosen strategies where no agent unilaterally wishes to change its choice. In MAS, some knowledge about the other agents should be accessible when simulating the outcome of strategies. This knowledge (e.g., the

payoff matrix of another agent, and the knowledge that it maximizes its expected utility) makes it hard to predict the outcome of the actual conflict. Instead of having a single prediction we end up with allowing almost any strategy. This is a consequence of the so-called Folk Theorem (see, e.g., [29, 59],).

A game can be modeled as a strategic or an extensive game. The former is a model of a situation in which each agent choose a plan of action once and for all, and all agents' decisions are made simultaneously while the latter specifies the possible orders of events. All the agents in this paper use strategic strategies, which we classify as generous, even-matched, or greedy. An interesting analogy can be made between this methodological choice and the advocating of pluralism with respect to the selection of choice rules for more advanced utility-based agents [15, 59]. In section 4.3 the outcomes for 15 different strategies are shown as an example of our classification.

Secondly the ESS can be described as a collection of successful strategies, given a population of different strategies. An ESS is a strategy in the sense that if all the members of a population adopt it, then no mutant strategy can invade the population under the influence of natural selection. A successful strategy is one that dominates the population, therefore it will tend to meet copies of itself. Conversely, if it is not successful against copies of itself, it will not dominate the population. In an evolutionary context, we can therefore simply calculate how successful an agent will be. The problem is that this is not the same as finding a successful strategy in an iterated game because in this game the agents are supposed to know the history of the moves.

Instead of finding the best one, we can try to find a possibly sub-optimal but robust strategy in a specific environment, and this strategy may eventually be an ESS. If the given collection of strategies is allowed to compete over generations (population tournament), we will eventually find a winner, but not necessarily the same one for every repetition of the game. In section 4.4 a round robin tournament is held for prisoner's dilemma like games to see what kind of strategy that will do best and population tournaments illustrate what successful combinations there are.

Thirdly the ESS can be seen as a collection of evolved successful strategies. It is possible to simulate a game through a process of two crucial steps: mutation (changes in the ways agents act) and selection (choice of the preferred strategies). Different kinds of evolutionary computations (see e.g., [36, 51]) have been applied within the MAS society, but the similarities to biology are restricted.¹ In section 4.5 we introduce noise and the agents become uncertain about the out-

¹Firstly, evolutionary computations, use a fitness function instead of using dominating and recessive genes. Secondly, there is a crossover between parents instead of the meiotic crossover.

come of the game, even if they have complete knowledge about the context.

4.2 Prisoner's dilemma like games

Prisoner's dilemma was originally formulated as a paradox (in the sense of that of Allais and Ellsberg) where the cooperatively preferable solution for both agents, low punishment, was not chosen. The reason is that the first agent did not know what the second agent intended to do, so he had to guard himself. The paradox lies in the fact that both agents had to accept a high penalty in spite of that cooperation is a better solution for both of them [60, 72].

The one-turn prisoner's dilemma has one dominant strategy, play defect. If the game is iterated there will be other dominating strategies because the agents have background information about previous moves. The Iterated Prisoner's Dilemma (IPD) is generally viewed as the major game-theoretical paradigm for the evolution of cooperation based on reciprocity.

When Axelrod and Hamilton analyzed the iterated prisoner's dilemma they found that the cooperating *TfT* strategy did very well against more defecting strategies [6, 8]. All agents are interested in maximizing individual utilities and are not pre-disposed to help each other. If an agent cooperates this is not because of an undirected altruism but because of a reciprocal altruism favoring a selfish agent [87]. The *TfT* strategy has become an informal guiding principle for reciprocal altruism [3, 4].

A *TfT*-agent begins with cooperation and then imitates the other agent in a game lasting an unknown number of times. A defecting agent will always win when meeting a *TfT* agent. In spite of that, a group of *TfT* agents will be stable against invasion of agents using other strategies because they are doing well when meeting their own strategy. No other strategy can do better against *TfT* than the strategy itself. Depending on the surroundings this will be the best strategy, as in Axelrod's simulations, or a marginally acceptable or even a bad strategy.

Binmore gives a critical review of the *TfT* strategy and of Axelrod's simulation [13]. He concludes that *TfT* is only one of a very large number of equilibrium strategies and that *TfT* is not evolutionary stable. On the other hand evolutionary pressures will tend to select equilibrium for the IPD in which the agents cooperate in the long run. In the next section we will look at an alternative interpretation.

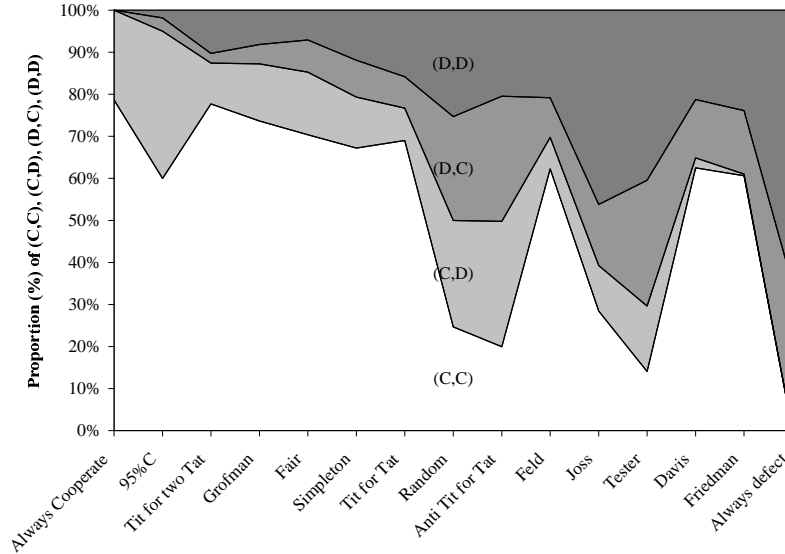


Figure 4.1: Proportions of (C,C), (C,D), (D,C) and (D,D) for different strategies.

4.3 A simulation example

In a simulation we used the proportions of (C,C), (C,D), (D,C) and (D,D) to analyze the successfulness of a strategy. We have developed a simulation tool (see Appendix A) in which we let 15 different strategies meet each other. The different strategies are described in table 3.2 at page 44.

The tournament was conducted in a round robin way so that each strategy was paired with each other strategy plus its own twin and a play random strategy. Each game in the tournament was played on average 100 times (randomly stopped) and repeated 5000 times. The outcomes are shown in figure 4.1 below where the percentage of (C,C), (C,D), (D,C) and (D,D) for each strategy is shown. We will use the proportions of (C,C), (C,D), (D,C) and (D,D) as "fingerprints" for the strategy in the given environment, independent of the payoff ma-

trix. For some of the strategies this is true without any doubts: Always Cooperate (*AllC*) and Always Defect (*AllD*) have 100 per cent cooperate $(C,C) + (C,D)$ and 100 per cent defect $(D,C) + (D,D)$ respectively. It is possible to look at how the proportions of (C,D) compared to (D,C) form different groups of strategies. *TfT* begins with cooperate and then does the same move as the other player did last time. This means that $(C,D) \approx (D,C)$ for all payoff matrices so the actual values do not matter. It is possible to treat the other strategies the same way because none of them reflect upon their actual payoff value. We will instead describe the strategies as *generous*, *even-matched* or *greedy*.

1. A *generous strategy* cooperates more than its partners do. This means that $(C,D) > (D,C)$ i.e. it is betrayed more often than it plays defection against a cooperate agent itself.
2. An *even-matched strategy* has $(C,D) \approx (D,C)$. This group includes the *TfT* strategy, always doing the same as the other strategy.
3. A *greedy strategy* defects more than its partners do. This means that $(C,D) < (D,C)$, i.e., the opposite to a generous strategy.

The basis of the subdivision above is a zero-sum play. The sum of the strategies (C,D) must equal the sum of the strategies (D,C) , i.e., if there is a generous strategy there must also be a greedy strategy. The classification of a strategy can change depending on the surrounding strategies. Theoretically a lot of changes are possible making a generous strategy become an even-matched or a greedy strategy, or doing it in a reverse order. What will happen with a particular strategy depends both of the surrounding and the character of the strategy. As an example *AllC* will always be generous while 95%C will change to a greedy strategy when there are only these two strategies left.

4.4 Simulating four different games

Assume that we have the following matrix, 4.1, for a general game where C and D is the strategic choices the two players have to make. As can be seen the letters k, l, m and n are the payoffs for (C,C) , (C,D) , (D,C) and (D,D) respectively in a symmetric game. The average payoff for a strategy $E_{avg}(strategy)$ is a function of the payoff matrix and the distribution of the payoffs among the four outcomes.

$$E_{avg}(strategy) = p(C,C)k + p(C,D)l + p(D,C)m + p(D,D)n, \quad (4.1)$$

$$\text{where } p(C,C) + p(C,D) + p(D,C) + p(D,D) = 1 \quad (4.2)$$

	C	D
C	k, k	l, m
D	m, l	n, n

Table 4.1: A basic 2x2 matrix

Prisoner's dilemma			Chicken Game			Coordination Game			Compr. Dilemma		
	C	D		C	D		C	D		C	D
C	3,3	0,4	C	3,3	1,4	C	2,2	0,0	C	2,2	2,3
D	4,0	1,1	D	4,1	0,0	D	0,0	1,1	D	3,2	1,1

Table 4.2: Payoff matrices for prisoner's dilemma, chicken game, coordination game and compromise dilemma.

The aim of the simulation is to test how different games behave in a round robin tournament and in a population tournament. We used four different games, prisoner's dilemma (PD), chicken game, coordination game and compromise dilemma games to illustrate the distributions among different strategies (see table 4.2). Additional information about the results of the simulations, definitions of the strategies, etc. can be found in [20]. It holds for all the games that (D,D) has a lower payoff value than (C,C) and for three of the games that (D,C) has the highest value. In an earlier paper we have examined the differences between PD and chicken game [21]. Compromise dilemma is closely related to chicken game. Coordination game is a game with two dominating strategies, playing (C,C) or playing (D,D). Rapoport and Guyer give a more detailed description of possible 2×2 games [73].

4.4.1 Round robin tournament

We ran a round robin tournament with the 15 strategies for the four different games described in figure 4.2.² The greedy strategies *Davis* and *Friedman* are doing well in PD while chicken game and coordinate game favor the generous strategies *AllC* and *Fair* respectively *Tf2T*. Compromise dilemma favored the

²For a full description of the strategies, see table 3.2 on page 44 [20].

counter intuitive strategy *ATfT*. In our classification *TfT* is regarded as an even-matched strategy. There is no reason for believing PD to favor more generous strategies than the rest of the games. Finding successful greedy strategies is well in line with the hypothesis that PD, because of the given payoff matrices, is the least cooperative game from the generous strategies point of view. The chicken game is less greedy than PD because it costs more to play defect (0 instead of 1 in the (D,D) case). The most successful strategies *AllC* and *Fair* are both generous. The coordinate game has the highest payoff value for (C,C), but it also has a dominating (D,D) value. The generous *Tf2T* is doing the best but the greedy strategies *Davis* and *Friedman* are also doing well compared to the other games. *Random* and *ATfT*, two strategies with a big proportion of (C,D) + (D,C) are doing very poorly in this game. In compromise dilemma (C,D) + (D,C) have high scores which favor the two even-matched strategies *Random* and *ATfT*.³ *ATfT* has the biggest proportion of (C,D) + (D,C) making it a winning strategy.

4.4.2 A population tournament

In a population tournament different strategies compete until there is only one strategy left or until the number of generations exceed 10.000. Because of changes in the distribution of strategies between different generations it is not possible to rely on previous descriptions of the strategies. A generous strategy can for example be greedy under certain circumstances. On average it must hold that there is the same amount of greedy strategies as generous ones, forming the even-matched strategies at the position of equilibrium. The population tournament was run 100 times for the four different games. It took between 2100 (compromise dilemma) and 3400 (chicken game) on average to find a winner in the game. At most a single strategy can win all the 100 times, but in our simulation different strategies won different runs. In all, five strategies were not winning a single game namely: *95%C*, *ATfT*, *Feld*, *Joss* and *Tester*. For the compromise dilemma, despite the fact that *ATfT* was the winner in the round robin tournament, the strategy did not win a single game in the population tournament. In the prisoners dilemma there is a change towards the originally more generous strategies *Tf2T* and *Grofman*. This is also true for the coordinate game, which also favors *AllC*, just as in the round robin tournament. For the chicken game the same generous strategies are doing well as in the PD and the coordinate game. The most surprising result is the almost total dominance of two greedy strategies, *Davis* and *Friedman* in compromise dilemma. Both strategies have a large proportion of (D,C) compared to (C,D) in the original round robin tournament. We

³*ATfT* does not have to be even-matched, it depends upon the actual surrounding.

also found the generous strategies to be more stable in the chicken game part of the matrix.

4.5 Noisy environment

In the next simulation, we introduced noise on four levels: 0.01, 0.1, 1 and 10%. This means that the strategies change to the opposite move for this given percentage.

In compromise dilemma *Friedman*, a greedy strategy dominates the population when the noise is 1% or below. *ATfT* is the second best strategy and together with *Fair* and *AllD* replace *Friedman* with 10% noise. Unlike the rest of the games there is a mixture of strategies winning each play for 0.1 to 10% noise.

Two greedy strategies are doing well in PD with none or a small level of noise. *Davis* is doing well without noise and *Friedman* with 0.01% noise. *Simpleton*, a generous strategy, is dominating the population when the noise is 0.1% or more.

In chicken game three generous strategies, *Tf2T*, *Grofman* and *Simpleton* are almost entirely dominating the population under noisy conditions. With increasing noise *Tf2T* first disappears then *Grofman* disappears leaving *Simpleton* as a single dominating strategy at 10% noise.

Finally in coordination game three generous strategies, *AllC*, *Tf2T* and *Grofman* are winning almost all the games when noise is introduced. With 10% noise *AllC* wins all the games.

4.6 Conclusions

We investigated four different PD like games in a round robin tournament and a population tournament. The results were analyzed using our classification of generous, even-matched and greedy strategies.

In the round robin tournament we found PD being the game which favored greedy strategy the most. The chicken game and the coordinate game were favoring generous strategies and compromise dilemma even-matched strategies. These results are not consistent with the common idea of treating the PD as the most important cooperating iterated game. We do not find these results surprising because all the used strategies are fully dependent on the mutual meetings.

The payoff of different games can easily be calculated using a linear function when the different proportions of (C,C), (C,D), (D,C) and (D,D) are known. If the game matrices are changed, different kind of strategies will be favored.

A more interesting investigation is to figure out what happens in a population tournament. If a strategy is generous, even-matched or greedy it is so only in

a particular surrounding and will possibly change when the strategies change. A winning strategy in a population tournament has to do well against itself because there will be lots of copies of that strategy. A winning strategy must also be good at resisting invasion from other competing strategies otherwise it will be impossible to become a single winner.

These restrictions in a population tournament make it natural to look for winning strategies among originally generous or even-matched (i.e. *TfT*) strategies. For three of the games, the PD, the chicken game and the coordination game, this is true with *Tf2T* and *Grofman* winning a big proportion of population games. Contrary to what was advocated by Axelrod and others, *TfT* was not among the most successful strategies.

The most divergent result was that compromise dilemma had two greedy strategies, *Davis* and *Friedman*, almost entirely dominating the population tournament. Both *Davis* and *Friedman* are favoring playing defect against a cooperate agent but unlike *AllD* they are also able to play cooperate against a cooperate agent. Despite a close relationship to the PD, the compromise dilemma finds other, more greedy, successful strategies.

When noise was introduced to the games, chicken game and coordinating game almost entirely favored generous strategies. In PD and even more in compromise dilemma the greedy, *Friedman* strategy was doing well.

For prisoner's dilemma, chicken game and coordination game the number of successful strategies decrease when noises are introduced. Equilibrium consisting of a lot of strategies is replaced by one to four dominating strategies. In both chicken game and coordination game these strategies (*Simpleton*, *AllC*, *Grofman* and *Tf2T*) are originally generous. In prisoner's dilemma the originally greedy strategy *Friedman* is also doing well with noise. For compromise dilemma there is a different situation with two greedy strategies, *Friedman* and *Davis*, that dominate without noise. With increasing noise a mixture of mostly greedy strategies is forming the winning concept.

We think these results can be explained by looking at the original game matrices. For chicken game (D,D) is doing the worst, favoring generous strategies. Coordination game gives (C,C) the highest results which out-scores greedy strategies. PD is, compared to chicken game, less punishing towards (D,D) which allows greedy strategies to become more successful. In compromise dilemma (C,D) and (D,C) have the best scores making a balance between different strategies possible.

Like ESS this description of MAS, as a competition between generous and greedy strategies, tries to find robust strategies that are able to resist invasion by other strategies. It is not possible to find a single best strategy that wins, but it is possible to tell what kinds of strategies which will be successful.

Chapter 5

Characteristic Distributions in Iterated Games

5.1 Introduction

We distinguish iterated games from repeated ones. The repeated games are games in which the players have no memory, while in the iterated games, the strategies remember all the previous actions that were made by the opponents, i.e. they have a history of the game so far.

Iterated strategic games are known to be harder to analyze and find equilibria in than repeated ones because of the exponentially increasing number of possible states (and thus taken in to consideration when choosing the next move). Recently, promising attempts have been made, especially in the field of evolutionary game theory, to use e.g. adaptive dynamics to describe how equilibria might be reached among simple strategies in iterated games [40, 80, 91]. While these attempts try to answer the question *“How are strategies behaving?”*, we will here try to focus on the question *“What is the result of their behavior?”* and *“How can this result be used?”*. We have in previous chapters 3–4 [22, 47] briefly discussed the notion of Characteristic Distributions (*ChDs* or “fingerprints”) to describe strategies in certain environments and will elaborate this concept a bit further here. Figure 5.1 on page 62 gives an overview of how the *ChDs* are created by the strategies and how the agents may use them in order to choose strategies. Previous papers discussing meta-games (i.e. the game of choosing a strategy for a game) include the work by Binmore and Samuelson [14], Abreu and Rubinstein [1], and Rubinstein [77].

First, we describe the distinction between agents and strategies and cover some formalities (section 5.2). A simple example will be given in section 5.3,

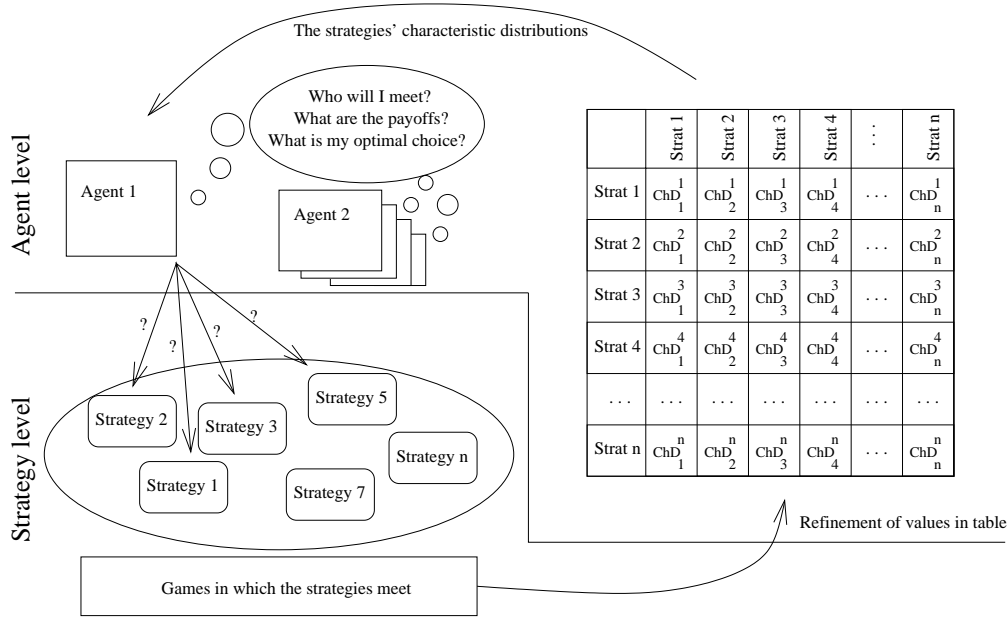


Figure 5.1: The two-layered approach to games. The different agents choose strategies possibly based on their predictions about the other agents' choice of strategies and what game is being played.

and the following section presents the following three theorems on optimality in games:

1. All meta-games of choosing strategies for games have Nash equilibria.
2. For each strategy and each opponent setup, there are games for which the setup is optimal.
3. Taken over all games, all strategies are equally good ("*No Free Lunch Theorem*" for game theory)

Finally some conclusions and further work are presented in section 5.5.

5.2 Characteristic Distributions - Definitions

To present the ideas in a somewhat formal and clear way, we need to define some central concepts such as strategies, games and *ChDs*.

Definition 1 (Strategy) *By a strategy we mean a function that projects the sequence of previous actions of itself and its opponents to the set of possible actions.*

The actual language used to define the strategies does *not* matter to us, i.e. it is irrelevant if the strategies are described by Moore or Mealy automata [42], by a programming language such as C [63], by bit strings [56], or even a mixture of descriptions. As long as the strategies are able to play a game (as defined below), possibly given a history of the game so far, they will fit our definition.

Definition 2 (Game) *A game is a situation where each combination of the choice of actions of n strategies is projected to a value in D^n , where D is the domain of values.*

The definition states that for each move, each participating strategy is assigned a certain value (the payoff of the move). From here on, only strategic¹, symmetric games are considered, unless explicitly said otherwise. The ideas may easily be extended to asymmetric games.

Definition 3 (Size of a Game) *The size d_G of a game G is the number of possible combinations of actions in each iteration.*

Remark 1 The size of a game with n players where each strategy have k actions to choose from is k^n .

For example a two-player two-choice game has size 4, a two-player asymmetric game in which one player have three choices and the other four choices is of size 12, a three-player four-choice symmetric game has size 64, etc.²

Definition 4 (Agent) *An agent is a meta-strategy that choose strategies for playing games.*

This general definition does not cover all aspects of agency, e.g. rationality, reactivity, abilities to communicate or to model other agents beliefs, desires and intentions.³ Instead, we focus on their role as selectors of ways to behave, i.e. choosing strategies. As we will see, characteristic distributions provide information to the agent of what each choice of strategy will pay. *How* this choice is done by the agents is not treated in this paper.

¹In strategic games, all strategies make synchronous choice of actions.

²Often, games such as the n -person Prisoners Dilemma are described as games of size $2(n + 1)$, but this is actually a special case in which outcomes with the same number of cooperators are grouped together, regardless of *who* cooperated.

³An overview of these properties (and more) can be found in [96].

Definition 5 (Population) A population (of strategies), P is the union of the sets of strategies that the agents consider in a particular game.

There are two things worth noting here. Firstly, the agents may or may not consider choosing from the same set of strategies. In the case where there are strategies that an agent does not know of, the probability of meeting such a strategy is set to 0 (for that agent). Secondly, note that we distinguish between the agent level and the strategy level, where the strategies are purely projective, while the agents may have capabilities to reason about other agents choices of strategies, to decide what game is the most suitable for describing the present situation, etc.

Definition 6 (Characteristic Distribution) The characteristic distribution (ChD) of a strategy s , when meeting another strategy t , denoted ChD_t^s for a game of size d is the d -entry matrix that tells the distribution of outcomes (i.e. combinations of moves made by the strategies) when strategy s , meets the strategy t . $ChD_t^s(i)$ is the i :th entry in that matrix.

The enumeration of the entries is reduced to one index i , although the ChD -matrix have the same dimension as the number of players and a normal indexation would need as many index variables as there are dimensions. This simplification is valid, as long as the enumeration of the entries is unambiguous.

Remark 2 Since all possible outcomes are considered, the sum of the entries, i.e. $\sum_{i=1}^{d_G} ChD_t^s(i) = 1$, for all $s, t \in P$.

Remark 3 For two-player games, $ChD_t^s = (ChD_s^t)^T$.

Definition 7 (Population Distribution) The population distribution, P_d , of a population P , is the function $P_d : P \rightarrow [0, 1]$ that tells the estimated probability of meeting each of the strategies in the population; especially, let P_d^a denote the population distribution function of agent a .

Remark 4 Since P_d is a probability distribution, it has the property of summing up to 1, i.e. $\sum_{t \in P} P_d(t) = 1$, since all considered strategies are in P .

Remark 5 There may be a difference between the actual distribution of strategies in the population – P_d , and the distribution agent a is considering – P_d^a , since what strategy a choose will affect P_d while it will not affect the expected opponent strategies P_d^a .

Definition 8 (Weighted Characteristic Distribution) We let $\hat{ChD}_{P_d}^s$ denote the weighted ChD , i.e. the sum $\sum_{t \in P} P_d^s(t) \cdot ChD_t^s$.

Lemma 1 $\sum_{i=1}^{d_G} ChD_{P_d}^s(i) = 1$

For a proof, see appendix C.

Definition 9 (Payoff Matrix) *The payoff matrix of a game G of size d , denoted π_G is a matrix of real numbers of size d . Let $\pi_G(i)$ be the i :th entry of the matrix.*

The payoff matrix is, since it assign each combination of actions payoffs, of the same dimensions as the $ChDs$ and the same enumeration is used to point out the entries.

Definition 10 (Payoff) *Let s be a strategy. Its payoff $\pi_G(ChD_t^s)$ in a game G when meeting an opponent strategy t is defined by*

$$\pi_G(ChD_t^s) = \sum_{i=1}^{d_G} \pi_G(i) \cdot ChD_t^s(i) \quad (5.1)$$

Since the payoff simply is a linear function of the $ChDs$, it is easy to determine what strategy is the most successful one in a certain environment of other strategies. It is also easy to take a subset of the entries in a ChD and make comparisons between them. An example of such a comparison is the one done in chapter 3 where we studied two of these derivated properties, - generosity and greediness [22].⁴ We showed that generous strategies got higher payoffs in noisy chicken games than the greedy ones.

Of course, we can pick other subsets of the entries in the $ChDs$, give them appropriate attributes and compare them between the strategies, but that is beyond the scope of the current work. Also, we will not treat the problematic issue of how the agents model their opponent agents and their choices of strategies. Instead we argue that the $ChDs$ are basis for making a good choice of strategy in iterated strategic symmetric games.

5.3 Characteristic Distributions - an Example

To make it easier to see how the $ChDs$ work, we will look into an example, starting with a clarification of the distinction (as we see it) between *strategies* and *agents* in iterated games.

⁴A *generous* strategy is a strategy that cooperates more than its opponent does, i.e. the proportion of CD is greater than that of DC, opposed to the *greedy* strategy. For a full definition of these properties, see section 3.3 [22].

$ChD_{s_2}^{s_1}$			$ChD_{s_1}^{s_2}$		
	a_1	a_2		a_1	a_2
a_1	0.4129	0.3282	a_1	0.4129	0.1904
a_2	0.1904	0.0685	a_2	0.3282	0.0685

Table 5.1: Characteristic Distributions of the meeting between s_1 and s_2

5.3.1 The two-layered approach to games

We consider agents as players in a meta-game of choosing right strategies for the actual game. This approach have been discussed by e.g. Binmore and Samuelson [14], Abreu and Rubinstein [1] and Rubinstein [77].

We will treat *strategies* as being simple automata, choosing the next moves strictly on what they know about the state of the game so far. Further, *agents*, are the actors trying to model the environment (possibly including other agents and the context of the game) and based on what it knows, choose an appropriate strategy by means of the *ChDs*, see figure 5.1.

As proposed in chapters 2 through 4 [22][47], the result of the interaction between two strategies is not dependent on what game is being played at the moment. The strategies do by definition not take the payoff matrix into consideration. Instead, such decisions are left to the agent. This pragmatic perspective clarify the role of the strategies in our theory and it will turn out to be an extremely helpful point of view.

5.3.2 An Example

Consider the following simple example as an illustration of how *ChDs* may be used. We have a world consisting of four different strategies, i.e. $P = \{s_1, \dots, s_4\}$. When two agents play a two-choice game, each of the agents use one of the four strategies (say, s_1 and s_2). The result of their choices of strategy for the game can be described by two matrices, one for each agent. These matrices consist of the probability distributions of the different outcomes of the game and thus the sum of the entries is 1 (see figure 5.1).

How are then these numbers computed? To answer that question, we must bear in mind that the only input to the strategies is the actions of the previous iterations. In other words: there are no *other* environmental factors that can have an influence on the *ChD* than the strategies themselves, which makes it possi-

	a_1	a_2
a_1	3	0
a_2	4	1

Table 5.2: A payoff matrix for a prisoners dilemma

ble to determine the values to arbitrary accuracy. The values may be calculated analytically or estimated through computer simulations. Or, we can simply look them up in a database, given that the strategies are known to the agent and the result of them playing is saved.

Values corresponding to those in table 5.1 can be made for the other pairs of strategies, including strategies meeting themselves and $\hat{C}hDs$ in the case where the agent has a distribution function for the strategies in the population. An example: If the agent predicts that the probabilities that the other agents choose strategies s_1 through s_4 are 0.17, 0.04, 0.58 and 0.21 respectively, the $\hat{C}hD$ for e.g. s_1 , $- \hat{C}hD_{P_d}^{s_1}$ is:

$$0.17 \cdot ChD_{s_1}^{s_1} + 0.04 \cdot ChD_{s_2}^{s_1} + 0.58 \cdot ChD_{s_3}^{s_1} + 0.21 \cdot ChD_{s_4}^{s_1} \quad (5.2)$$

Note that this way of setting the probabilities makes it possible to use $\hat{C}hDs$ in situations where the agents have different impact on the probabilities. Agents that one agent have a greater probability of interaction, e.g. in a situated game⁵, or in a Multi Agent System (MAS) in which coalitions of agents increase the probability of interaction.

As mentioned in section 5.1, the *result* of the meeting between two strategies is independent of what game is played. The result of the *game* however, is the sum of the elements of the result matrix, multiplied by the corresponding entries in the payoff matrix, see definition 10.

If we for instance say that the game played was a prisoners dilemma with a payoff matrix as in 5.2, the expected payoff for s_1 would be:

$$0.4129 \cdot 3 + 0.3282 \cdot 0 + 0.1904 \cdot 4 + 0.0685 \cdot 1 = 2.0688 \quad (5.3)$$

and s_2 would probably score:

$$0.4129 \cdot 3 + 0.1904 \cdot 0 + 0.3282 \cdot 4 + 0.0685 \cdot 1 = 2.6200 \quad (5.4)$$

⁵An example of a situated game is a game in which the players are placed on e.g. a grid, making each of them having local environments in which they act [57].

Another payoff matrix will of course result in different scores for the strategies.

In what follows, we will show that for every possible environment, if the opponent strategies are known (with regard to both the probability of meeting them in the population and their *ChDs*), it will be possible to either choose a strategy from the present set of strategies considered, or construct a mixed strategy from these strategies, that is optimal.⁶ Nash showed in 1950 that this was the case for mixed strategies in repeated games [67] and we will show that the result is also applicable for iterated games using the *ChDs*.

5.4 Optimality in games

Based on the discussion above, we will now draw some conclusions from the point of view of both the game and the *ChD*. First, we conclude that it is always possible to find an optimal strategy or a mix of strategies when the environment is given.

Secondly, we show that it is always possible to construct a game that makes a strategy in a population optimal. To find a game in which a strategy is strictly optimal for a certain population distribution is harder and in fact it is not possible for all cases.

5.4.1 Optimal strategies in a game

Theorem 2 *Given a population of strategies P (able of playing an arbitrary strategic game G), the choice of a mix of strategies has a Nash equilibrium.*

Sketch of proof: (a full proof is given in appendix C):

We conclude that the choice of strategies fulfills the conditions of Kakutani's fixed point theorem and therefore the choice of an optimal strategy must have a fixed point, which is a Nash equilibrium. \square

The result as such mean that we, regardless of what game we play, know that there is a distribution of strategies that is (at least locally) optimal, even if the underlying game, for which we choose the strategies, may lack such equilibria.

5.4.2 Optimal games for a strategy

On the one hand, we have proved that it is always possible to find equilibria of optimal strategies in a game. On the other hand, we may as well prove the possibility of finding an optimal *game* for a certain strategy.

⁶We do by an *optimal strategy* mean the strategy that has the highest payoff.

Theorem 3 *For all strategies s and a population distribution P_d ,*

1. *It is always possible to find a game G in which s is optimal.*
2. *If $ChD_{P_d}^s$ is a corner of the convex hull of the set of $ChDs$, it is always possible to find a strictly optimal game for s .*

For the proof, see appendix C.

The implication of this result is that we may *not* tell generally that a certain strategy, in a certain environment will be unsuitable for all games. On the contrary, we may always find games in which every strategy, given an arbitrary, but specifically chosen distribution of opponents, will be among the best.

5.4.3 “No Free Lunch Theorem” for strategies

Theorem 4 *Let G be the set of all possible games. Then, for arbitrary strategies s_1 and s_2 and population distributions P_d^1, P_d^2 :*

$$\sum_G \pi_G(ChD_{P_d^1}^{s_1}) = \sum_G \pi_G(ChD_{P_d^2}^{s_2}) \quad (5.5)$$

A proof is given in appendix C.

This theorem states that no strategy is better than any other strategy, when all possible games are considered. We are therefore not able to tell whether a certain strategy is better than any other strategy, without being provided by context-specific information.

5.5 Conclusions and Future Work

We have introduced the concept of Characteristic Distributions and explained how they can be used to structure knowledge about how different strategies behave when they meet. This knowledge is useful for agents in order to make optimal choices (in a given context). We also claim that it is always possible to find a strategy or a probability distribution of strategies that is optimal and that it is always possible to find a game for a strategy, in which it is optimal in a given environment of other strategies. Both these theorems were proven.

The $ChDs$ combined with the distributions of choices of strategies is a powerful tool for modeling an agents' choice of strategies. It may not only be used in round robin tournaments (where all meetings have the same impact on the result, i.e. P_d is constant), but in a variety of other settings such as:

1. Population dynamics in which the better strategies⁷ have a greater probability of surviving than the worse.
2. Different types of *situated games* where the choices of the closest neighbors (or the agents that are most probable that our agent will meet) will dominate the P_d .
3. Prevent invasions of “nasty” strategies in the population through asserting a class of such strategies a small probability in P_d . This will (if adopted by the agents in the system) prevent the choice of strategies that are *too* nice to the intruders, since their utility will be slightly lower than the more revenging strategies and thus decrease the risk of genetic drift.⁸

Some questions arise though, e.g. how does changes in the level of noise or the length of the game affect the theories? The theory works as long as they remain the same; however, if they are changed, the *ChD*'s will probably *not* change in some linear way.⁹

Another interesting aspect of these ideas is that they require that the agent is able to make a fairly good approximation of the probabilities of meeting other strategies, i.e. the choice of the other agents. The other agents' choices may, in turn, depend on their model of the first agents' choice, etc. Different methods are used to approximate these choices and this will be one of our future tasks to connect these ideas to the *ChDs*.

⁷If we by *the better* mean the ones that at the moment have the higher payoff.

⁸Genetic drift is a phenomenon that may occur when several strategies in a population have the same payoff, e.g. if the population only consists of *tit-for-tat* and *AllC* in a IPD without noise, there will be no difference in the payoff. This may lead to a situation where *AllC* take over the population although it is less fit to resist invasion from defecting agents than *Tit-for-Tat*

⁹This claim is endorsed by the critique of Axelrod's Prisoners Dilemma tournaments [6], where e.g. Bendor shows how relatively small increases in the level of noise lead to decreases in the payoff for Tit-for-Tat [11].

Chapter 6

Conclusions and Future Work

"To look backward for a while, is to restore the eye and render it more fit for its prime object of looking forward."

Margaret Fairless Barber

6.1 Conclusions

In a chapter like this, one is supposed to present the main contributions of the work, – what makes it worth the time and effort it takes to read this licentiate thesis?¹ During the work with the text there have been a few insights that have characterized the points of view taken:

1. Strategies are automata. Basically, we treat strategies as automata which are totally unaware of what payoff matrix is valid. They are also unaware of the level of noise in the game, how many iterations the game is played, etc.
2. Agents choose strategies. We treat agents as being the *smarter ones*, who may have abilities to calculate probabilities that other agents will choose a certain strategy, and to estimate a payoff matrix and other attributes in a game.
3. General games are more interesting than specific ones. There is, as we see it, no good reason to believe that a one payoff matrix should be of

¹Except for my dear colleagues and family, who of course will never admit that they have not read it, on the other hand they will get a copy of it whether they want to or not. I guess it is a matter of pathetic self-realization from my side that I would like to see something that I have written in someone else's bookshelf that makes me give it to people whose shelves I am likely to pass now and then. But name that researcher who does *not* have such feelings :-)

greater interest than any other.² Therefore we have tried to look upon parameterized games, as well as we have defined some somewhat more general measures (s.a. the notions of generous/greedy strategies) instead of cheering (or, for that sake dishonor) a certain strategy that do well (or bad) in a certain game.

4. Noise is an important factor. We believe that, even though artificial agents are deterministic to a higher extent than e.g. people are, there are still noisy situations. Communication lines may be unstable, programs may be buggy and an operating system may go down.³

Starting with these ways of looking on the subject, we have concluded the following:

1. Chicken game is, in some situations, a game that better describes the actual payoff for the players (the ambulance example of page 28 is one example). Unfortunately, it has been put into the shade by the prisoner's dilemma, but we think that it should be given more attention as the use of game theory in MAS technology mature.
2. The payoff of a strategy is highly dependent on two things: what game is played, and what opponents does it face. A strategy may do well in some environment, but may be easily outperformed in another and this makes it impossible to draw any general conclusions of how a strategy will perform. This statement is supported in practice by simulations (chapter 2-4 [21, 22, 47]) and in theory by the theorems of chapter 5.
3. No strategy is better than another, taken over all possible games. This is a result of the fact that all strategies have the same payoff, if all possible games are summed together (the "*No Free Lunch Theorem*" for strategies).
4. The notion of population distributions and characteristic distributions may prevent genetic drift. By never letting the probability of invasion of nasty strategies in the population distribution reach zero (even though there is no risk of invasion), genetic drift will be prevented. The small fraction of virtual intruders in the population distribution works as a differentiator between the strategies that are open to and those that are resistant to invasion.

²At least not from the point of view of the artificial agent, who is to choose a strategy for the game it predicts will be played. For scientists, as we all know, some types of games seem to be more interesting than others.

³I do not point out any specific operating system that bugs out several times a day running applications that they have written themselves, due to the risk of getting sued by some rich American.

5. Preliminary simulations have showed to support the idea of mixed equilibria in the meta-games of choosing strategies for iterated, non-cooperative strategic games.⁴
6. Generosity and Greediness are two (of many) possible classifications of strategies.⁵ In chapters 3 and 4 we showed that generosity generally pays in chicken games. However, the properties are dependent on the opponents. As an example, in population games, the population distributions affect the weighted *ChDs* so that the more successful strategies will have more impact on it than the worse ones. A result of this is that when a strategy is taking over the population, it will meet more copies of itself, and as in the case of our classification, the strategy will become more and more even-matched, i.e. have about equal shares of the generous (C,D) moves and the greedy (D,C) moves. All winning strategies of population games must be even-matched.⁶

In all, the *ChD*-notion and its pragmatic point of view to game theory, - to look at the result of strategies playing, instead of trying to analyze why a certain strategies made certain actions, is both powerful and useful. Its generality in its approach to the games, the possibility to protect against genetic drift, and the clear distinctions between the agent layer and the strategy layer provide a suitable framework for agent encounters.

6.2 Future Work

The area of game theory and its use in multi agent systems is doubtless an interesting and, as we see it, promising area. We believe that *ChDs* and the point of view of regarding agents as selectors of behaviors may open up the area and make the way for the use of game theoretic tools in real world MAS problems.

⁴This have been done through simulations of convergence phenomena of populations with equal individuals, i.e., simulation of genetic drift and comparing those to simulations with several strategies. The populations with several strategies did not converge (in noncooperative iterative games), as opposed to the populations of the former, thus empirically showing the existence of the equilibria.

⁵We only consider the *outcomes* of the whole games, i.e. comparisons between entries in the *ChDs*.

⁶In more general terms, all winners of population games must be balanced in the diagonal of their *ChD*, having as large proportion of outcomes beneath it as above it. This is a direct result of the fact that it is a zero-sum game at that level. What is lost for one strategy, is won by another, and since both "sides of the move" belong to the same strategy, it will be registered at both sides of the diagonal in the *ChD* matrix.

A continuation of the theoretic work is one possible trajectory. This would include developing the “*No Free Lunch Theorem*” for strategies presented in chapter 5 and its implications, looking at criteria for optima in games and to prove some (unpublished) empirical results theoretically⁷.

Another interesting piece of work is to evaluate the usefulness of *ChDs* in the valuation between agents and coalitions. A (boundedly) rational agent should weigh advantages and drawbacks in its decisions on joining and defecting coalitions in its environment. At the same time, the coalitions (or the rest of the agents in the coalition) make the same judgments about the individual agents.

⁷For instance, what is the connection between payoff matrices, *ChDs*, the number of individuals in a population, and the number of iterations before convergence?

Appendix A

Simulation Tool for Strategies

A.1 Introduction

SITS is a tool for simulating games between strategies in symmetric two-strategy, two-choice games. It was developed in the spring of 1997 as a subproject of GLOSS¹ by the authors. It is written in C and the current version 1.0 β seems to be stable on the Solaris platform, but bug reports are still very welcome, and so are any suggestions on how to improve the implementation, preferably by e-mail to the author.²

A.2 The Strategy Language

The language used for programming strategies in SITS is a LISP-like language in which expressions belonging to it always evaluates to an action. It also contains registers, local to each strategy, which can be modified. The language is described in detail in the forthcoming subsections.

A.2.1 The Syntax of a Strategy

A strategy must have three parts in order to be syntactically (and semantically) correct.

1. A string containing its name.
2. An expression that, when evaluated, returns its start action.

¹The Group of Large Open SystemS, a subdivision of SOC, the Societies of Computation research group at the University of Karlskrona/Ronneby, Sweden

²`sja@ipd.hk-r.se`

3. An expression that, when evaluated, returns the action of the strategy (for the rest of the actions).

The strategy must, in the strategy file, be on (at least) a row of its own and its parts are separated by ‘;’s. A strategy ends by a ‘!’.

Example 4 (A definition of a strategy)

`Tit for tat; C; ifoppC(C,D)!`

A.2.2 Terminals

The *terminals* take no arguments and they all (except the integer-terminal) have C or D as a value.

C C is an atom in the language whose value is C, i.e. the cooperate move.

D D is an atom in the language whose value is D i.e. the defect move.

Real numbers

A real number in the range of $[0, 1]$ will be interpreted as a D with the probability of the given number, else C.³

r1 The real number **r1** will be interpreted as a D with a probability given of the value of the register **r1**, else C.³

cr1 The complementary action of the **r1**-terminal. Returns C whenever **r1** would return D and D whenever **r1** would return C.

r2, cr2

These terminals are analogous to **r1** and **cr1**.

Integers

Integers are arguments to some special operators and are interpreted as usual. They are not defined as values of a strategy tree, but serve as arguments to some non-terminals, e.g. `ifnoCgtn`.

A.2.3 Non-terminals

Non-terminals, or *operators*, take at least one argument. Some argument may be numeric and are then used in comparisons with registers etc. in order to determine the branch of evaluation.

³Every time this kind of terminal is evaluated, a new random number $r \in [0, 1]$ is generated and compared to the terminal. If it is greater than the terminal, it evaluates to C, else it will evaluate to D.

ifoppC(expr1, expr2)

ifoppC returns the value of **expr1** if the last action of the opponent was C, else it returns the value of **expr2**. Consecutive use of **ifoppC** will look at the second, third, fourth,... last actions of the opponent. The history of actions is restored when the evaluation of the strategy is finished.

ifIC(expr1, expr2)

ifIC returns the value of **expr1** if the last action own action was C, else it returns the value of **expr2**. Consecutive use of **ifIC** will look at the second, third, fourth,... last own action. The history of actions is restored when the evaluation is finished.

prog2(expr1, expr2)

prog2 is a sequencer that first evaluates **expr1** and then evaluates **expr2**. The value of the **prog2**-expression is the value of evaluating **expr2**.

resetr1(expr1)

resetr1 sets the value of **r1** to 0 and returns the value of evaluating **expr1**.

resetr2(expr1)

resetr2 sets the value of **r2** to 0 and returns the value of evaluating **expr1**.

incrr1(expr1, float)

incrr1 first increases **r1** with the value of the second argument, then evaluates and returns the value of **expr1**.

multr1(expr1, float)

multr1 first multiplies **r1** with the value of the second argument, then evaluates and returns the value of **expr1**.

incrr2(expr1, int)

incrr2 first increases **r2** with the value of the second argument, then evaluates and returns the value of **expr1**.

ifr1gtr2(expr1, expr2)

If **r1**>**r2**, the operator returns the value of evaluating **expr1**, else it returns the value of evaluating **expr2**.

ifr2gtr1(expr1, expr2)

ifr2gtr1 returns the value of evaluating **expr1** if **r2**>**r1**, else it returns the value of evaluating **expr2**.

`ifr1gtz(expr1, expr2)`

`ifr1gtz` returns the value of evaluating `expr1` if the value of evaluating `r1` > 0 else it returns the value of evaluating `expr2`.

`ifr1gtm(expr1, expr2, int)`

`ifr1gtm` returns the value of evaluating `expr1` if the value of evaluating `r1` > the value of evaluating the third argument, `int`, else it returns the value of evaluating `expr2`.

`ifr2gtm(expr1, expr2, int)`

`ifr2gtm` returns the value of evaluating `expr1` if the value of evaluating `r2` > the value of evaluating the third argument, `int`, else it returns the value of evaluating `expr2`.

`ifnoCgtm(expr1, expr2, int)`

`ifnoCgtm` returns the value of evaluating `expr1` if the number of cooperations done by the opponent⁴ exceeds the value of evaluating the third argument.

`ifitgtm(expr1, expr2, int)`

`ifitgtm` returns the value of evaluating its first branch, `expr1`, if the current iteration is greater than the integer `int` given as its last argument. Otherwise it returns the value of evaluating its second argument.

A.2.4 BNF grammar for the Strategies

A BNF grammar is given in table A.1 for the language we are using to define the strategies in.

Table A.1: A BNF grammar for the strategy language

<code><strategy> ::=</code>	<code><string> ; <expr> ; <expr> !</code>
<code><string> ::=</code>	<code><character> <character><string></code>
<code><character> ::=</code>	<code>a b c d e f g h i j k l m n o p q r</code> <code> s t u v x y z A B C D E F G H I J </code> <code>K L M N O P Q R S T U V X Y Z 1 2 3</code> <code> 4 5 6 7 8 9 0 ! " % & / - +</code>
<code><expr> ::=</code>	<code><terminal> <non-terminal></code>
<code><terminal> ::=</code>	<code>C D r1 cr1 r2 cr2 <probability></code>

⁴That is, the number of cooperative moves done by the opponent and stored in the history.

Table A.1: A BNF grammar for the strategy language

<code><probability> ::=</code>	<code>0.<digits> 1.0</code>
<code><digits> ::=</code>	<code><digit> <digit><digits></code>
<code><digit> ::=</code>	<code>0 1 2 3 4 5 6 7 8 9</code>
<code><non-terminal> ::=</code>	<code><unary-op> <binary-op> <ternary-op></code>
<code><unary-op> ::=</code>	<code>resetr1(<expr>) resetr2(<expr>)</code>
<code><binary-op> ::=</code>	<code>incrr1(<expr>,<real-number>) </code> <code>incrr2(<expr>,<real-number>) </code> <code>multr1(<expr>,<real-number>) </code> <code>multr2(<expr>,<real-number>) </code> <code>ifr1gtr2(<expr>,<expr>) </code> <code>ifr2gtr1(<expr>,<expr>) </code> <code>ifr1gtz(<expr>,<expr>) </code> <code>prog2(<expr>,<expr>) </code> <code>ifIC(<expr>,<expr>) </code> <code>ifoppC(<expr>,<expr>)</code>
<code><ternary-op> ::=</code>	<code>ifnoCgtn(<expr>,<expr>,<digits>) </code> <code>ifitgtn(<expr>,<expr>,<digits>) </code> <code>ifr1gtn(<expr>,<expr>,<digits>) </code> <code>ifr2gtn(<expr>,<expr>,<digits>)</code>
<code><real-number> ::=</code>	<code><digits>.<digits></code>

A.2.5 The strategy file

The file containing the strategies must have the following properties

1. It must start with the row
Number of Strategies = x
where x is the number of strategies that you want to include in the simulation.
2. A number of strategies, as described in the previous sections. There must be at least as many strategies as you declare in the initial row (see above).

Here is an example of a strategy file:

Example 5 (An example of a strategy file with 5 strategies of which 4 are used.)

```

Number of strategies = 4
Moren; C; ifoppC(C,0.3333333)!
Persson; C; ifitgtn(ifIC(D,C),ifoppC(D,C),1)!
Gardenfors; 0.5;
    prog2(prog2(resetr1(N),
        prog2(prog2(ifIC(incr1(N,1),N),
            ifIC(incr1(N,1),N)),
            prog2(ifIC(incr1(N,1),N),
                ifIC(incr1(N,1),N)))),
        ifr1gtn(0.1,ifr1gtn(0.7,ifr1gtn(0.5, 0.7, 0), 1), 2))!
Random; 0.5; 0.5!
AllD; D; D!

```

As can be seen, the last of the strategies, **AllD**, will not be part of the simulation, since the number of strategies in the initial row is set to 4 and only the first four strategies are considered.

A.3 Running the SITS system

The SITS system have been implemented to run on the SOLARIS platform. Efforts on porting the system to other platforms have been initialized, but are still unfinished.

A.3.1 How to configure SITS

In the SITS directory is a file named **config.h**. This file contains most of the parameters that can be used to configure the simulations.

STRATEGY_FILE

STRATEGY_FILE is the name of the file from where the strategies to the simulation is read. Its full address must be given.

EXPECTED_ITERATIONS

This is the number of expected iterations, or actions, in each game. The actual number of iterations played in a game is exponentially distributed, making it impossible for a strategy to take advantage of when a game will end, since the probability for the game to end in each iteration is the same

	C	D
C	R,R	S,T
D	T,S	P,P

Table A.2: The Payoff matrix for a symmetric, two-strategy, two-choice game

N_O_GAMES

The number of games between every chosen pair of strategies (each with a number of iterations). The more meetings, the longer the simulation will take, but also, the more accurate the result will be.

NOISE The level of noise in the simulation in % of the actions.⁵ 50 corresponds to randomness, 100 to the anti-strategies, 1 to 1% of noise.

The Payoff matrix

The payoff matrix is described in the table below. In the configuration file, the values in the matrix is set by assigning values to the parameters **R_VALUE**, **S_VALUE**, **T_VALUE** and **P_VALUE**, corresponding to R, S, T and P in the table, respectively. The payoff matrix used in the games will then look like in table A.2 above.

SIM_TYPE

This parameter choose the type of simulation to be run. At the moment, only one type of simulation is implemented, that is the **ROUND_ROBIN**.

EVOLUTIONARY

This parameter turns on the evolutionary mode when set to 1 (0 turn it off). When in the evolutionary mode, SITS first runs an ordinary tournament, then a population dynamics simulation is run for a number of generations.

N_O_GENERATIONS

This parameter sets the maximum number of generations in the evolutionary game.

POPULATION_SIZE

POPULATION_SIZE sets the size of the population in the evolutionary game.

⁵The noise used in the simulations is of the *trembling hand* type, i.e. both of the players see what action was made and their histories remains consistent.

OUTPUT If set to 1, SITS will prints a trace of how long the simulation has proceeded to the standard output.

PRINT_PLAYERS

When set to 1 SITS will dump the strategies source code to the chosen log output]

PRINT_MEETINGS

When **PRINT_MEETINGS** is set to 1 it prints the result of each meeting to the chosen log output.

PRINT_GENERATIONS

This parameter will print the distribution of strategies in each generation to the chosen log output when set to 1.

LOG_FILE

This is the name of the file to where the output will be written. Setting it to **NULL** will direct the output to the standard output (normally the window where SITS is run).⁶

LOG_FILE_MODE

The output may be written to the output file either by overwriting the previous contents of the file, or by adding it to the end. **a** appends logged data at the end of the file, while **w** overwrites existing data. The **a** option is very useful when several runs of SITS are to be done and the results of them is to be collected in one file.

A.3.2 How to start SITS

When all parameters are set in the **config.h** file, the SITS system needs to be recompiled. This is done by writing **comp** [**return**] at the prompt starting the compilation script.

After the compilation, you will have a newly created executable file named **SITS** in your directory. Writing **SITS** [**return**] will start the simulation. This may take a while, depending on the size of the simulation. You can count on an approximated 10.000 iterations (each containing the evaluation of two strategy trees) per second, and 100.000 evaluated individuals per second (in the population game).⁷

⁶If it is set to **NULL**, the trace of the simulations set by the **OUTPUT** parameter will not work

⁷The performance is measured on a Sun SPARCstation 4 running Solaris

A.3.3 How to run series of simulations

When you want to run series of simulations, e.g. you may want to study the result of varying both noise and the number of expected iterations, the easiest way is to use a shell script to run it for you.

Create a new file that will contain your script (an example of a script that varies two parameters is given in example 6).

After you have written your script, you must change its rights, so that it will be executable. This is done by typing (again, we assume the name of the script to be `newsim`) `chmod 700 newsim` in the directory you compiled it in.

Example 6 (An example of a script that varies `P_VALUE` and `NOISE`)

```
# script to test foreach
foreach i (0 0.1 0.2 0.3 0.4 0.5)
  foreach j (0 0.01 0.03 0.1 0.3 1.0)
    gcc -lm -o sits -DP_VALUE=$i -DNOISE=$j
      runtest.c simulation.c game.c player.c
      lexer.c strategy.c history.c
    echo "P=$i, noise=$j%"
    SITS
  end
end
echo "finished"
```

In this example two parameters are varied, `P_VALUE` and `NOISE` and it may serve as an example of how to such scripts. The `foreach` is an iterator that assigns the variable each of the values within the parenthesis and then executes the commands until it reaches its `end`.

The `gcc -lm -o sits -DP_VALUE=$i -DNOISE=$j runtest.c ...` is the call to the compiler `gcc` to compile the given c-files into a file named `SITS`. The `-D` flag sets the constant following it to the new value. In this case, `-DP_VALUE=$i` means that the `P_VALUE` is assigned the value of the variable `i` (which is taken from the list `(0 0.1 0.2 ...)` in the `foreach` command).

Caution! You must edit the `config.h`-file in such manner that the parameters that you assign new values in the script does not get their old values back in the `config.h`-file. I.e. you must comment out the settings of that parameters in the `config.h`-file. This is either done by commenting out the definition, or by adding `ifndef PARAMETER` before and `endif` after the row defining the parameter (`PARAMETER` should be exchanged for the name of the parameter you would like to vary).

Also, you should not forget to set the `LOG_FILE_MODE` to `a` if you use a log-file. Else you will only get the result of the last run in your series logged.

Appendix B

Implementation of the strategies in SITS

For a description of the functionality of the strategies, see table 3.2.

```
Grofman; C; ifIC(ifoppC(C,0.71),  
                ifoppC(0.71,C))!
```

```
TfT; C; ifoppC(C,D)!
```

```
Simpleton; C; ifIC(ifoppC(C,D),  
                  ifoppC(D,C))!
```

```
Random;0.5;0.5!
```

```
Friedman; C; ifIC(ifoppC(C,D,)   
                  D)!
```

```
All C; C; C!
```

```
Davis; C; ifitgtn(ifoppC(ifIC(C,D),  
                        D),  
                  C,  
                  10)!
```

```
Tf2T; C; ifoppC(C,  
                ifoppC(C,D))!
```

ATfT; D; ifoppC(D,C)!

Fair; C; ifr1gtr2(ifoppC(resetr1(C),D),
 ifr2gtr1(ifoppC(resetr2(C),C),
 ifIC(ifoppC(C,incrr1(D,1)),
 ifoppC(incrr2(C,1),C))))!

95%C; 0.05; 0.05!

Feld; resetr1(C); ifitgtn(ifoppC(0.5,D),
 incrr1(ifoppC(r1,D),
 0.0025),
 200)!

Joss; C; ifoppC(0.1,D)!

Tester; D; ifr1gtr2(ifoppC(C,D),
 ifoppC(ifIC(D,C),
 incrr1(D,1)))!

All D; D; D!

Appendix C

Proofs

Lemma 1

$$\sum_{i=1}^{d_G} ChD_{P_d}^s(i) = 1$$

Proof 1

$$\begin{aligned} \sum_{i=1}^{d_G} ChD_{P_d}^s(i) &= \sum_{i=1}^{d_G} \sum_{t \in P} P_d(t) \cdot ChD_t^s(i) = \sum_{t \in P} P_d(t) \sum_{i=1}^{d_G} ChD_t^s(i) \text{ (Remark 2)} \\ &= \sum_{t \in P} P_d(t) \cdot 1 \text{ (Remark 4)} = 1 \end{aligned} \tag{C.1}$$

Theorem 2

Given a population of strategies P , able of playing an arbitrary iterated strategic game G , the choice of a mix of strategies has a Nash equilibrium.

Proof 2 ¹ Let $P_m \ni P_d$ be the set of all mixed strategies² of P , and let $r : P_d \Rightarrow P_m$ be the relation from the population distribution to the distribution of strategies that maximize the payoff for the agent. A fixed point in r is a set of choices ρ such that $\rho \subseteq r(\rho)$. It is easy to realize that for all P_d , r is a non-empty relation, i.e. there is at least one best response to each distribution of opponent

¹Modified version of Theorem 1.1 in [30], pp 29–30

²The term “mixed strategy” is here used on the meta-level and describes the probabilities of the agent to choose a certain strategy in the game. These strategies may, or may not, in themselves be mixed strategies in the underlying game.

strategies.³ To show that the choice of strategy have a Nash equilibrium it is sufficient to show the existence of a fixed point in the game:

Kakutani's fixed point theorem[48] shows that for $r : P_d \rightrightarrows P_m$ to have a fixed point, it is sufficient that the following conditions are fulfilled:

1. P_d is a compact, convex and nonempty subset of a (finite-dimensional) Euclidean space
2. $r(\rho)$ is nonempty for all ρ .
3. $r(\rho)$ is convex for all ρ .
4. $r(\cdot)$ has a closed graph: If the sequence of optimal strategies⁴ $(\rho^n, \hat{\rho}^n) \rightarrow (\rho, \hat{\rho})$ with $\hat{\rho}^n \in r(\rho^n)$, then $\hat{\rho} \in r(\rho)$ (upper hemi-continuity).

Condition 1: P_d is a vector of real numbers in $[0, 1]$ of dimension $(|P| - 1)$ which is a compact, convex and non-empty subset of an Euclidean space. \square

Condition 2: Each choice P_m of strategies by an agent is continuous since it may distribute its choices arbitrarily over the set of strategies. Continuous functions on compact sets have maxima. Therefore there must be at least one $r(\rho)$. \square

Condition 3: (proof by contradiction) If $r(\rho)$ were not convex, there would be a $\rho' \in r(\rho)$ and a $\rho'' \in r(\rho)$ such that a $\lambda \in [0, 1]$ would fulfill $\lambda\rho' + (1 - \lambda)\rho'' \notin r(\rho)$. However, for all agents playing ρ' in $\lambda\%$ of the cases and ρ'' the remaining cases, the following will hold:

$$\pi_G(\lambda ChD_{P_d}^{\rho'} + (1 - \lambda)ChD_{P_d}^{\rho''}) = \pi_G(\lambda ChD_{P_d}^{\rho'}) + (1 - \lambda)\pi_G(ChD_{P_d}^{\rho''}) \quad (C.2)$$

So if ρ', ρ'' both are in $r(\rho)$, then so must their weighted average (since they both have the same optimal payoff), which concludes the proof of condition 3. \square

Condition 4: Assume the contrary, i.e. that there is a sequence of choices of optimal strategies $(\rho^n, \hat{\rho}^n) \rightarrow (\rho, \hat{\rho})$ with $\hat{\rho}^n \in r(\rho^n)$, but $\hat{\rho} \notin r(\rho)$.

Then there must be one agent, i , for whom $\hat{\rho}_i \notin r_i(\rho)$ and a $\delta > 0$ such that for a ρ' we have $\pi_G(ChD_{P_d}^{\rho'}) > \pi_G(ChD_{P_d}^{\hat{\rho}}) + 3\delta$. Since π_G is continuous and $(\rho^n, \hat{\rho}^n) \rightarrow (\rho, \hat{\rho})$ (for sufficiently large n), we have:

$$\pi_G(ChD_{P_d}^{\rho'}) > \pi_G(ChD_{P_d}^{\rho'}) - \delta > \pi_G(ChD_{P_d}^{\hat{\rho}}) + 2\delta > \pi_G(ChD_{P_d}^{\hat{\rho}^n}) + \delta \quad (C.3)$$

Thus ρ' will do strictly better than $\hat{\rho}^n$ against ρ^n which contradicts that $\rho^n \in r(\rho^n)$. Therefore the assumption that $\hat{\rho} \notin r(\rho)$ must be false and the condition

³We may simply compare the payoffs of the strategies we choose from and take the one that pays off best.

⁴ (x, \hat{x}) denotes that my best action was x as a response to the rest of the population choosing \hat{x} .

true.

We have now show that G has fixed points and since fixed points of r are Nash equilibria, this concludes the whole proof. \square

Theorem 3

For all strategies s and a population distribution P_d ,

1. it is always possible to find a game G in which s is optimal.
2. if $ChD_{P_d}^s$ is a corner of the convex hull of all weighted $ChDs$, it is always possible to find an strictly optimal game for s .

Proof 3 The first claim is easily proven. All games in which $\forall i, j \in [1, d_G](\pi_G(i) = \pi_G(j))$ fulfills the condition by assigning all strategies the same score and thus making them equally good/bad. \square

The second claim is shown by for a strategy s , find such a game G and then prove that the properties of s being a corner are sufficient for it to be strictly optimal in G .

The definition of corners in $ChDs$ tells us that for a strategy s to be a corner, there must be a combination of entries $ChD^s : [1, \dots, d_G] \rightarrow \{0, 1\}$ that s has the largest proportion of in the expected population P of strategies. Call the set of entries with a non-zero ChD^s value C_s . We know that for all strategies $t \in P$, $\sum_{i \in C_s} ChD^s > \sum_{i \in C_s} ChD^t$. Now consider the game

$$\pi_G(i) = \begin{cases} 0 & i \notin C_s, \\ 1 & i \in C_s. \end{cases} \quad (C.4)$$

The payoff for a strategy s in this game is equal to its proportion of the entries that it has the most of according to the assumption, $\sum_{i \in C_s} ChD^s(i)$ and thus it has to be optimal. \square

Theorem 4

For all strategies s_1 and s_2 and population distributions P_d^1 and P_d^2 ,

$$\sum_G \pi_G(ChD_{P_d^1}^{s_1}) = \sum_G \pi_G(ChD_{P_d^2}^{s_2}) \quad (C.5)$$

Proof 4 The idea of the proof is similar to the one of Wolpert and Macready [94], where they show that for two arbitrary search algorithms, the sum of their performance taken over all possible goal functions, is equal. We will show that for two arbitrarily chosen strategies in a population of strategies, their payoff, taken over all possible payoff matrices, is equal, i.e.

$$\sum_G \pi_G(ChD_{P_d^1}^{s_1}) = \sum_G \pi_G(ChD_{P_d^2}^{s_2}) \quad (C.6)$$

Let π_G be the payoff function of game G . A strategy can be said to outperform another strategy, if it have a higher payoff in the game played and our claim is that the number of games in which one is better than the other equals the number of cases in which the latter is better than the former. From the definition of the payoff function⁵ inserted in the equation above follows:

$$\begin{aligned} \sum_G \pi_G(ChD_{P_d}^s) &= \sum_G \sum_{i=1}^{d_G} \pi_G(i) ChD_{P_d}^s(i) = \\ &= \sum_{i=1}^{d_G} \sum_{\pi_G(i) \in D} \pi_G(i) ChD_{P_d}^s(i) \end{aligned} \quad (C.7)$$

Since the $ChD_{P_d}^s$ is independent of the payoff, we may remove it outside the sum of payoffs which leaves us with:

$$\sum_{i=1}^{d_G} ChD_{P_d}^s(i) \cdot \sum_{\pi_G(i) \in D} \pi_G(i) \quad (C.8)$$

The inner sum is independent of i because it describes the sum of all payoffs for G and i only decides the order of summation, which is irrelevant to the result. Call the result of that inner summation ζ . We get:

$$\sum_{i=1}^{d_G} ChD_{P_d}^s(i) \cdot \zeta = \zeta \cdot \sum_{i=1}^{d_G} ChD_{P_d}^s(i) (\text{Lemma 1}) = \zeta \cdot 1 = \zeta \quad (C.9)$$

Due to lemma 1, we then have that it all equal ζ and thus the payoff taken over all possible games is independent not only of what strategy we choose, but also of what opponents it face.⁶ \square

⁵Definition 10 at page 65.

⁶ ζ is typically equal to 0 when summing all games $\in \mathfrak{R}^{d_G}$

Bibliography

- [1] D. Abreu and A. Rubinstein. The structure of nash equilibrium in repeated games with finite automata. *Econometrica*, 56:1259–1282, 1988.
- [2] R.J. Aumann, M.B. Maschler, and Stearns R.E. *Repeated Games with Incomplete Information*. MIT Press, 1995.
- [3] R. Axelrod. Effective choice in the prisoner’s dilemma. *Journal of Conflict Resolution*, 24(1):379–403, 1980.
- [4] R. Axelrod. More effective choice in the prisoner’s dilemma. *Journal of Conflict Resolution*, 24(3):3–25, 1980.
- [5] R. Axelrod. The emergence of cooperation among egoists. *The American Political Science Review*, 75:306–318, 1981.
- [6] R. Axelrod. *The Evolution of Cooperation*. Basic Books Inc., 1984.
- [7] R. Axelrod and D. Dion. The further evolution of cooperation. *Nature*, 242:1385–1390, 1988.
- [8] R. Axelrod and Hamilton W.D. The evolution of cooperation. *Science*, 211, 1981.
- [9] D.H. Ballard. *An Introduction to Natural Computation*. MIT Press, 1997.
- [10] J. Bendor. Uncertainty and the evolution of cooperation. *Journal of Conflict resolution*, 37(4):709–734, 1993.
- [11] J. Bendor, R.M. Kramer, and Stout S. When in doubt... cooperation in a noisy prisoner’s dilemma. *Journal of conflict resolution*, 35(4):691–719, 1991.
- [12] J. Bendor and P. Swistak. The evolutionary advantage of conditional cooperation. *Complexity*, 4(2):15–18, 1998.
- [13] K. Binmore. *Playing Fair: game theory and the social contract*. MIT Press, 1994.
- [14] K. Binmore and L. Samuelson. Evolutionary stability in repeated games played by finite automata. *Journal of Economic Theory*, 57:278–305, 1992.
- [15] M. Boman and L. Ekenberg. Decision making agents with relatively unbounded rationality. In Nawarecki, editor, *Proceedings of DIMAS’95*, pages 28–35, 1995.

- [16] M. Bonatti, Y.M. Ermoliev, and A.A. Gaivoronski. Modeling of multi-agent systems in the presence of uncertainty: The case of information economy. Technical report, IIASA Working Paper WP-96-94, 1996.
- [17] R. Boyd and J.P. Lorberbaum. No pure strategy is evolutionary stable in the repeated prisoners dilemma game. *Nature*, 327:58–59, 1987.
- [18] R.A. Brooks. Elephants dont play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.
- [19] K. Carley and A. Newell. The nature of the social agent. *Journal of Mathematical Sociology*, 19(4):221–262, 1994.
- [20] B. Carlsson. Evolutionary models in multi-agent systems, 1998. Licentiate thesis.
- [21] B. Carlsson and S.J. Johansson. An iterated hawk-and-dove game. In W. Wobcke, M. Pagnucco, and C. Zhang, editors, *Agents and Multi-Agent Systems*, volume 1441 of *Lecture Notes in Artificial Intelligence*, pages 25–37. Springer Verlag, 1998.
- [22] B. Carlsson, S.J. Johansson, and M. Boman. Generous and greedy strategies. In *Proceedings of Complex Systems 98*, 1998.
- [23] C. Castelfranchi and Conte R. Limits of economic and strategic rationality for agents and multi-agent systems. *Robotics and Autonomous Systems*, 24(3-4):127–139, 1998.
- [24] R. Conte and E. (eds) Chattoe. *Evolving Societies: the computer simulation of social systems*. Sage, 1998.
- [25] E.H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.
- [26] J.M. Epstein. Zones of cooperation in demographic prisoner’s dilemma. *Complexity*, 4(2):36–48, 1998.
- [27] J.M. Epstein and R. Axtell. *Growing Artificial Societies. Social Science from the Bottom Up*. MIT Press, 1996.
- [28] J. Farrell and R Ware. Evolutionary stability in the repeated prisoners dilemma. *Theoretical Population Biology*, 36:161–166, 1988.
- [29] D. Fudenberg and E. Maskin. The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, 80(2):274–279, 1986.
- [30] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.
- [31] P. Gärdenfors. *Game Theory and Ethics*, pages 194–197. Bokförlaget Nya Doxa, 1981.
- [32] P. Gärdenfors. *Blotta tanken*. Bokförlaget Nya Doxa, 1992.
- [33] D. Gauthier. *Morals by Agreement*. Oxford University Press, 1985.
- [34] M.R. Genesereth, M.L. Ginsberg, and J.S. Rosenschein. Cooperation without communication. In Bond and Gasser, editors, *Distributed Artificial Intelligence*, pages 220–226. Kaufmann, 1988.

- [35] M.R. Genesereth and S.P. Ketchpel. Software agents. *Communications of the ACM*, 37(7):48–53, 1994.
- [36] D.E. Goldberg. *Genetic Algorithms*. Addison-Wesley, 1989.
- [37] M. Goossens, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1994.
- [38] R. Gustavsson. Multi agent systems as open societies – a design framework. In *Intelligent Agents IV, Agent Theories, Architectures and Languages*, volume 1365 of *Lecture Notes in Artificial Intelligence*, pages 29–37. Springer Verlag, 1998.
- [39] S. Hägg. A sentinel approach to fault handling in multi-agent systems. In *Proceedings of the Fourth Pacific Rim International Conference on Artificial Intelligence, PRICAI-96, Lecture Notes in Artificial Intelligence*, 1114. Springer Verlag, 1996.
- [40] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, 1998.
- [41] D.R. Hofstadter. *Metamagical Themas: Questing for the Essence of Mind and Pattern*. Basic Books, HarperCollins Publishers, 1985.
- [42] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading MA, 1979.
- [43] M.N. Huhns and L.M. Stephens. Multi-agent systems and societies of agents. In WeißG., editor, *Multiagent Systems - A modern approach to distributed artificial intelligence*, pages 79–120. MIT Press, 1999.
- [44] N.R. Jennings. Agent-based computing: Promise and perils. In *Proceedings of IJCAI 99*, 1999. The paper of the talk of the IJCAI Computers and Thought Award.
- [45] S.J. Johansson. Mutual valuations between agents and their coalitions. to be presented at *Intelligent Agent Technology - 99*, Dec. 1999, 1999.
- [46] S.J. Johansson and B. Carlsson. Mini manual for the use of sits, simulation tool for strategies. Manuscript, available on request.
- [47] S.J. Johansson, B. Carlsson, and M. Boman. Modelling strategies as generous and greedy in prisoners dilemma like games. In *Proceedings of SEAL 98*, 1998.
- [48] S. Kakutani. A generalization of brouwer’s fixed point theorem. *Duke Mathematical Journal*, 8:457–459, 1941.
- [49] D. Koller and A. Pfeffer. Generating and solving imperfect information games. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) '95*, pages 1185–1192, 1995.
- [50] D. Koller and A. Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215, 1997.
- [51] J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

- [52] D. Kraines and V. Kraines. Pavlov and the prisoner's dilemma. *Theory and Decision*, 26:47–79, 1989.
- [53] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence Journal*, 94(1-2):79–98, 1997.
- [54] S. Kraus, K. Sycara, and Evenchik A. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence Journal*, 104(1-2):1–69, 1998.
- [55] L. Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, 1994.
- [56] K. Lindgren. Evolutionary phenomena in simple dynamics. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial life II*. Addison Wesley, 1991.
- [57] K. Lindgren. Evolutionary dynamics in game-theory models. In W.B. Arthur, D. Lane, and S.N. Durlauf, editors, *The Economy as an Evolving, Complex System II*, pages 337–367. Addison Wesley, 1997.
- [58] J. Linnerooth-Bayer, L. MacKellar, and K. Dake. Risk and fairness - local and global issues, 1996. Presented at the Colloquium "Umweltgerechtes Handeln: Motivation und Barrieren".
- [59] B. Lomborg. Game theory vs. multiple agents: The iterated prisoner's dilemma. In C. Castelfranchi and E. Werner, editors, *Artificial Social Systems*, volume 830 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 1994.
- [60] R.D. Luce and H. Raiffa. *Games and Decisions*. Dover Publications, 1957.
- [61] D. MacLean. Variations on fairness. In *Proceedings of the Workshop on Risk and Fairness*, IIASA, Laxenberg, Austria, 1996.
- [62] L. Marinoff. The inapplicability of evolutionarily stable strategy to the prisoner's dilemma. *Brittish Journal of Philosophical Science*, 41:461–472, 1990.
- [63] P. Mathieu and J.P. Delahaye. Home page of a simulation tool for the iterated prisoner's dilemma. <http://www.lifl.fr/mathieu/ipd/>.
- [64] J. Maynard Smith. *Evolution and the theory of games*. Cambridge University Press, 1982.
- [65] J. Maynard Smith and G.R. Price. The logic of animal conflict. *Nature*, 246, 1973.
- [66] P. Molander. The optimal level of generosity in a selfish uncertain environment. *Journal of Conflict Resolution*, 29(4):611–618, 1985.
- [67] J. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences*, volume 36, pages 48–49, 1950.
- [68] M. Nowak and K. Sigmund. The evolution of stochastic strategies in the prisoner's, dilemma. *Acta Applicandae Mathematicae*, 20:247–265, 1990.

- [69] J. Pitt and A. Mamdani. A protocol-based semantics for an agent communication language. In *Proceedings of IJCAI 99*, 1999.
- [70] W. Rabinowicz. Grappling with the centipede. Technical report, Department of Philosophy, Lund University, 1996.
- [71] H. Raiffa. *Decision Analysis, Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, 1968.
- [72] A. Rapoport and A.M. Chammah. *Prisoner's Dilemma A Study in Conflict and Cooperation*. The University of Michigan Press, 1965.
- [73] A. Rapoport and M. Guyer. A taxonomy of 2×2 games. In *Yearbook of the Society for General Systems Research*, volume 11, pages 203–214. Society for General Systems Research, 1966.
- [74] E. Rasmusen. *Games and Information*. Blackwell, second edition, 1994.
- [75] E. Rich and K. Knight. *Artificial Intelligence*. McGraw-Hill, 1991.
- [76] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.
- [77] A. Rubinstein. Finite automata play the repeated prisoners dilemma. *Journal of Economic Theory*, 39:83–96, 1986.
- [78] B. Russell. *Common Sense and Nuclear Warfare*. Simon and Schuster, 1959.
- [79] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.
- [80] L. Samuelson. *Equilibrium Selection and Evolutionary Games*. MIT Press, 1997.
- [81] T.W. Sandholm and R.H. Crites. Multiagent reinforcement learning in the iterated prisoners dilemma. *BioSystems*, 37:147–166, 1996.
- [82] T.W. Sandholm and F. Ygge. On the gains and losses of speculation in equilibrium markets. In *Proceedings of IJCAI 1997*, 1997.
- [83] R. Schuessler. The gradual decline of cooperation: Endgame effects in evolutionary game theory. *Theory and Decision*, 26:133–155, 1989.
- [84] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.
- [85] R. Selten. The chain store paradox. *Theory and Decision*, 9:127–159, 1978.
- [86] R.G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transaction on Computers*, C-29(12):1104–1113, 1980.
- [87] R.L. Trivers. The evolution of reciprocal altruism. *Quarterly Review of Biology*, 46:35–57, 1971.
- [88] K.G. Troitzsch, U. Mueller, N. Gilbert, and J. (eds) Doran. *Social Science Microsimulation*. Springer Verlag, 1996.

- [89] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley and Sons, New York, 1944.
- [90] S.N.K. Watt. Artificial societies and psychological agents. In Nwana H.S. and Azarmi, editors, *Software Agents and Soft Computing - Toward Enhancing Machine Intelligence - Concepts and Applications*, volume 1198 of *Lecture Notes in Computer Science*, pages 27–42. Springer Verlag, 1997.
- [91] J. Weibull. *Evolutionary Game Theory*. MIT Press, 1996.
- [92] G. Weiß. *Multiagent Systems, - a modern approach to distributed artificial intelligence*. MIT Press, 1999.
- [93] M.A. Wellman. A computational market model for distributed configuration design. In *Proceedings of AAAI '94*. Morgan-Kaufman, 1994.
- [94] D.H. Wolpert and W.G. Macready. No free lunch theorems for search. Technical report, Santa Fe Institute, 1996. SFI-TR-95-02-010.
- [95] M. Wooldridge. Agent-based software engineering. *IEE Proceedings on Software Engineering*, 144:26–37, 1997.
- [96] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1994.
- [97] F. Ygge. *Market-Oriented Programming and its Application to Power Load Management*. PhD thesis, University of Lund, Department of Computing and Numerical Analysis, 1998.

Author Index

- Abreu D., 13, 61, 66
 Aumann R.J., 9
 Axelrod R., 10, 11, 24, 30, 31, 40, 43, 48, 53, 70
 Axtell R., 12

 Ballard, D.H., 13
 Bendor J., 11, 12, 48, 50, 70
 Binmore K., 41, 53, 61, 66
 Boman M., 19–21, 39, 42, 52, 56, 61, 66
 Bonatti M., 39
 Boyd R., 11
 Brooks R.A., 4

 Carley K., 4
 Carlsson B., 18–21, 42, 46, 56, 61, 66
 Castelfranchi C., 39
 Chammah A.M., 23, 26, 36, 53
 Conte R., 3, 39
 Crites R.H., 13

 Dake, K., 15
 Delahaye J.P., 31
 Dion D., 10, 11
 Durfee E.H., 2

 Ekenberg L., 39, 52
 Epstein J.M., 12
 Ermoliev Y.M., 39
 Evenchik A., 3

 Farrell J., 11
 Fudenberg D., 52

 Gaivoronski A.A., 39
 Gauthier, D., 15
 Genesereth M.R., 2, 3
 Goldberg D.E., 52
 Goosens M., 18
 Gustavsson R., 1
 Guyer M., 25, 56
 Gardenfors, P., 14, 15

 Hamilton W.D., 10, 24, 30, 40, 53
 Hofbauer J., 13, 61
 Hofstadter D.R., 9
 Hopcroft J., 63
 Hägg S., 3

 Jennings N., 1, 63
 Johansson S.J., 2, 3, 16, 18–21, 42, 43, 46, 56, 61, 66

 Kakutani S., 68, 88
 Ketchpel S.P., 2, 3
 Knight K., 5
 Koller D., 9
 Koza J.R., 52
 Kraines D., 11
 Kraines V., 11
 Kramer R.M., 11, 12, 50, 70
 Kraus S., 3

 Lamport L., 18
 Lindgren K., 12, 37, 44, 51, 63, 67
 Linnerooth-Bayer, J., 15
 Lomborg B., 24, 28, 39, 40, 48, 51, 52
 Lorberbaum J.P., 11
 Luce R.D., 8, 10, 23, 53

 MacKellar, L., 15
 MacLean, D., 15
 Macready W.G., 4
 Mamdani A., 2, 3
 Marinoff L., 11
 Maschler M.B., 9
 Mathieu P., 31
 Maynard Smith J., 13, 25, 27, 51
 Mittelbach F., 18
 Molander P., 11
 Morgenstern O., 39

 Nash J., 68
 von Neumann J., 39
 Newell A., 4
 Nordahl M., 12

Norvig P., 1, 5
Nowak, M., 13

Pfeffer A., 9
Pitt J., 2, 3
Price G.R., 25, 27

Rabinowicz W., 6
Raiffa H., 8, 10, 23, 53
Rapoport A., 10, 23, 25, 26, 36, 53, 56
Rasmusen E., 39
Rich E., 5
Rosenschein J.S., 14, 39, 51
Rubinstein A., 13, 61, 66
Russell B., 25
Russell S.J., 1, 5

Samarin A., 18
Samuelson L., 13, 61, 66
Sandholm T.W., 3, 13, 23
Schuessler R., 11
Selten R., 11
Sigmund K., 13, 61
Smith R.G., 3
Stearns R.E., 9
Stout S., 11, 12, 50, 70
Swistak P., 11
Sycara K., 3

Tirole J., 52
Trivers R.L., 30, 40, 53
Tucker, A.W., 10

Ullman J., 63

Ware R., 11
Watt S.N.K., 2
Weiß G., 3
Weibull J., 13, 61
Wellman M.A., 23, 51
Wolpert D.H., 4
Wooldridge M., 2, 63

Ygge F., 3, 23

Zlotkin G., 14, 39, 51

Subject Index

- 95%C, 45
 - definition of, 44
 - implementation of, 86
- adaptive dynamics, 13, 61
- agent
 - definition of
 - in game theory, 63
 - definitions of, 1–3
 - general description, 66
- agent coalition game
 - short description, 10
- agent component
 - in the social level, 4
- airplane landing example, 14
- AllC*, *see* always cooperate
- AllD*, *see* always defect
- always cooperate, 33, 36, 45, 55, 56
 - definition of, 44
 - implementation of, 85
- always defect, 33, 36, 45, 46, 48, 50, 55, 58
 - definition of, 44
 - implementation of, 86
- ambulance, 42
- anti-tit-for-tat, 45, 57, 58
 - definition of, 44
- applications of game theory, 13
- ATfT*
 - implementation of, 86
- ATfT*, *see* anti-tit-for-tat
- auctions, 3
- backward induction, 6
- battle of sexes, 8
- behavioral pattern, 1
 - BIBTEX, 18
- BNF grammar
 - for the strategy language, 78
- business matter example, 28
- C
 - SITS terminal, 76
- C 4 of 5, 35
- cardinal scale, 25
- centipede game, 5
- characteristic distribution, 50, 54
 - definition of, 64
 - example, 66–68
 - example of matrix, 66
 - weighted
 - definition of, 64
- ChD*, *see* characteristic distribution
- C \hat{h} D*
 - definition of, 64
- ChD*
 - definition of, 64
- C \hat{h} D*
 - example of, 67
- Chess, 9
- chicken game, 16, 36
 - description, 25
 - origin, 25
 - payoff matrix, 25, 56
- CO₂ emission, 15
- coalition game
 - description of, 6
 - payoff matrix, 7
- coalitions in MAS, 3
- company and customers, 8
- compromise dilemma
 - payoff matrix, 56
- computational market, 23
- config.h*
 - SITS input file, 80–82
- context component
 - in the social level, 3
- contract net, 3
- Coop puis tc strategy, 33
- coordination game
 - payoff matrix, 56
- coordination mechanism component
 - in the social level, 3

- cr1
 - SITS terminal, 76
- cr2
 - SITS terminal, 76
- creating script files for SITS, 83
- D
 - SITS terminal, 76
- Davis strategy, 45–48, 50, 56
 - implementation of, 85
- demiographic games, 12
- emission of CO₂
 - example, 15
- environment
 - noisy, 58
- even-matched strategy, *see* strategy→even-matched
- EVOLUTIONARY
 - SITS parameter, 81
- evolutionary computations, 52
- evolutionary game, 51
- evolutionary game theory, 13
- Evolutionary Stable Strategies, ESS, 51–53
- evolutionary stable strategy, 13
- EXPECTED_ITERATIONS
 - SITS parameter, 80
- extensive game, 52
- fair strategy, 45, 47, 48, 50, 56, 58
 - definition of, 44
 - implementation of, 86
- Feld strategy, 45, 57
 - definition of, 44
 - implementation of, 86
- fingerprint, *see* characteristic distribution
- fire brigade, 42
- flight boarding game, 9
- folk theorem, 36
- Friedman strategy, 45–47, 50, 56, 58
 - definition of, 44
 - implementation of, 85
- fuel example, 14
- game
 - definition of, 63
 - iterated, 24
 - situated, 70
 - size of
 - definition of, 63
- game theory
 - applications of, 13
 - role in multi agent systems, 39
- games
 - extensive, 5
 - favoring generous strategies, 59
 - favoring greedy strategies, 59
 - iterated vs. repeated, 61
 - payoff matrices, 45, 56
 - strategic, 5
- generosity, 65
- generous strategy, *see* strategy→generous
- gradual strategy, 33
- greediness, 65
- greedy strategy, *see* strategy→greedy
- Grofman strategy, 45, 47, 48, 57, 58
 - definition of, 44
 - implementation of, 85
- Hawk-and-Dove Game, 24
- hawk-and-dove game
 - description, 25
 - payoff matrix, 26, 27
- if IC
 - SITS operator, 77
- if itgtn
 - SITS operator, 78
- if noCgtn
 - SITS operator, 78
- if oppC
 - SITS operator, 77
- ifr1gtn
 - SITS operator, 78
- ifr1gtr2
 - SITS operator, 77
- ifr1gtz
 - SITS operator, 78
- ifr2gtn
 - SITS operator, 78

- `ifr2gtr1`
 - SITS operator, 77
- incomplete information in games, 8
- `incrr1`
 - SITS operator, 77
- `incrr2`
 - SITS operator, 77
- integer
 - sits terminal, 76
- invasion
 - prevention of, 70
- iterated games, 9
- Joss dur strategy, 33
- Joss mou strategy, 33
- Joss strategy, 45, 57
 - definition of, 44
 - implementation of, 86
- \LaTeX , 18
- `LOG_FILE`
 - SITS parameter, 82
- `LOG_FILE_MODE`
 - SITS parameter, 82
- market
 - computational, 23
- market game, 8
- Markov models, 13
- MaxAx
 - payoff matrix, 45
- meta-game, 66
- meta-strategy, 63
- MinAx
 - payoff matrix, 45
- misinterpretation noise, 11
 - example, 11
- multi-agent planning, 3
- `multr1`
 - SITS operator, 77
- mutation, 52
- `N_O_GAMES`
 - SITS parameter, 81
- `N_O_GENERATIONS`
 - SITS parameter, 81
- Nash equilibrium, 51
 - example, 28
- negotiation process, 23
- negotiations, 3
- NFL theorem, *see* “No Free Lunch” theorem
- “No Free Lunch”
 - theorem for strategies, 69
- “No Free Lunch”
 - theorem for search, 4
- NOISE
 - SITS parameter, 81
- noise
 - simulations with, 48
- noise in experiments, 58
- noisy environment, 58
- noisy games, 11
- ordinal payoff matrices, 25
- ordinal scale, 25
- Othello, 9
- outcomes
 - proportions of, 43
- OUTPUT
 - SITS parameter, 82
- output file
 - in SITS, 82
- P_VALUE
 - SITS parameter, 81
- Pavlov, *see* simpleton strategy
- payoff
 - average, 45
 - definition of, 65
 - example of, 67
- payoff matrix
 - SITS
 - definition of, 81
 - definition of, 65
- poker game, 9
- population
 - definition of, 64
- population distribution
 - definition of, 64
- population dynamics, 70

- population tournament, 47–48, 57
 - with noise, 48
 - without noise, 47
- POPULATION_SIZE
 - SITS parameter, 81
- PRINT_GENERATIONS
 - SITS parameter, 82
- PRINT_MEETINGS
 - SITS parameter, 82
- PRINT_PLAYERS
 - SITS parameter, 82
- prisoner's dilemma, 16
 - Axelrod's payoff matrix, 30
 - description, 23, 53
 - iterated, 53
 - payoff matrix, 25, 67
 - second condition, 26
 - symmetry in, 8
- prisoners dilemma
 - modified Axelrod payoff matrix, 45
 - payoff matrix, 56
- prog2
 - SITS operator, 77
- queuing game, 14
- r1
 - SITS terminal, 76
- r2
 - SITS terminal, 76
- R_VALUE
 - SITS parameter, 81
- random strategy, 35, 43, 45, 57
 - implementation of, 85
- real number
 - SITS terminal, 76
- reciprocal altruism, 53
- recursive strategy, *see* strategy→recursive
- repeated games, 9
- replicator dynamics, 13
- rescue vehicle, 42
- resetr1
 - SITS operator, 77
- resetr2
 - SITS operator, 77
- resource allocation
 - example, 28–29
 - payoff matrix, 45
 - with asymmetric restrictions, 8
- resource example, 43
- resource sharing example, 29
- right hand rule, 42
- rock, paper, scissors, 6
- roles in MAS, 3
- round robin, 54
- round robin tournament
 - probability distribution in, 69
- ROUND_ROBIN
 - SITS simulation type, 81
- S_VALUE
 - SITS parameter, 81
- scale
 - cardinal, 25
 - ordinal, 25
- script file
 - example of in SITS, 83
- selection, 52
- sentinel agents, 3
- sexes
 - battle of, 8
- singleton strategy, 45, 47, 48, 50, 58
 - definition of, 44
 - implementation of, 85
- SIM_TYPE
 - SITS parameter, 81
- simulation
 - setup, 43, 54
- simulations
 - series of, 83
- SITS
 - configuration, 80–82
- situated games, 12
- social conventions, 3
- society level
 - components of, 1
 - in MAS, 1–4
- spatial games, 12
- strategic game, 52
- strategies, *see* strategy

- definitions of, 44
- strategy
 - 95%C, *see* 95%C
 - always cooperate, *see* always cooperate
 - always defect, *see* always defect
 - anti-tit-for-tat, *see* anti-tit-for-tat
 - classification
 - example, 41
 - classification of, 41
 - Davis, *see* Davis strategy
 - definition of, 63
 - even-matched, 41, 55
 - evolutionary stable, 27, 35
 - examples, 27
 - properties, 27
 - fair, *see* fair strategy
 - Feld, *see* Feld strategy
 - Friedman, *see* Friedman strategy
 - general description, 66
 - generous, 41, 55, 65
 - greedy, 41, 55, 65
 - Grofman, *see* Grofman strategy
 - Joss, *see* Joss strategy
 - Pavlov, *see* simpleton strategy
 - properties, 41, 55
 - properties of a, 31, 36
 - random, *see* random strategy
 - recursive, *see* recursive strategy
 - simpleton, *see* simpleton strategy
 - stable, 27
 - syntax, 78
 - tester, *see* tester strategy
 - tit-for-2-tat, *see* tit-for-2-tat
 - tit-for-tat, *see* tit-for-tat
- strategy file
 - SITS syntax of, 79
- strategy language
 - BNF grammar, 78
- STRATEGY_FILE
 - SITS parameter, 80
- structure of a game, 3
- symmetry
 - payoff, 7
 - structural, 7
- table of examples, 8
- T_VALUE
 - SITS parameter, 81
- terminals
 - in SITS
 - definition of, 76
- territory defense
 - inverse strategy, 14
- tester strategy, 45, 57
 - definition of, 44
 - implementation of, 86
- Tf2T, *see* tit-for-2-tat
- Tf2T
 - implementation of, 85
- TfT, *see* tit-for-tat
- TfT strategy
 - implementation of, 85
- theorem
 - of “No Free Lunch”, 69
 - of optimal games, 69
 - of optimal strategies, 68
- theorem of optimal games
 - proof of, 89
- tit-for-2-tat, 45, 47, 56, 58
 - definition of, 44
- tit-for-tat, 33, 45, 47, 48, 50, 57
 - Axelrod’s opinion, 40, 53
 - critique, 41, 53, 70
 - definition of, 44
 - description, 30
- tournament
 - description, 54
 - population, 57
 - results, 56
- traffic intersection example, 29, 42–43
- trembling hand noise, 11
 - example, 11
- two-layered approach to games, 66
- urn guessing game, 9
- weighted characteristic distribution
 - example of, 67
- weighted *ChD*
 - definition of, 64