

Chapter 7

Simulating How to Cooperate in Iterated Chicken and Prisoner's Dilemma Games

Bengt Carlsson

Department of Software Engineering and Computer Science,
Blekinge Institute of Technology, Sweden

7.1 Introduction

In the field of multi-agent systems (MAS) the concept of game theory is widely in use ([15]; [23]; [30]). The initial aim of game theorists was to find principles of rational behavior. When an agent behaves rationally it “will act in order to achieve its goal and will not act in such a way as to prevent its goals from being achieved without good cause” [19]. In some situations it is rational to cooperate with other agents to achieve its goal. With the introduction of the “trembling hand” noise ([32]; [4]) a perfect strategy would take into account that agents occasionally do not perform the intended action¹. To learn, adapt, and evolve will be of a major interest for the agent. It became a major task for game theorists to describe the dynamical outcome of model games defined by strategies, payoffs, and adaptive mechanisms, rather than to prescribe solutions based on a priori reasoning. The crucial thing is what happens if the emphasis is on a conflict of interest among

¹ In this metaphor an agent choose between two buttons. The trembling hand may, by mistake, cause the agent to press the wrong button.

agents. How should in such situations agents cooperate with one another if at all?

A central assumption of classical game theory is that the agent will behave rationally and according to some criterion of self-interest. Most analyses of iterative cooperate games have focused on the payoff environment defined as the Prisoner's dilemma ([5]; [10]) while the similar chicken game to a much less extent has been analyzed. In this chapter, a large number of different (Prisoner's dilemma and chicken) games are analyzed for a limited number of simple strategies.

7.2 Background

Game theory tools have been primarily applied to human behavior, but have more recently been used for the design of automated interactions. Rosenschein and Zlotkin [30] give an example of two agents, each controlling a telecommunication network with associated resources such as communication lines, routing computers, short and long-term storage devices. The load that each agent has to handle varies over time, making it beneficial for each if they could share the resources, but not obvious for the common good. The interaction for coordinating these loads could involve prices for renting out resources within varying message traffic on each network. An agent may have its own goal trying to maximize its own profit.

In this chapter games with two agents each having two choices are considered². It is presumed that the different outcomes are measurable in terms of money or a time consuming value or something equivalent.

7.2.1 Prisoner's dilemma and chicken game

Prisoner's dilemma (PD) was originally formulated as a paradox where the obvious preferable solution for both prisoners, low punishment, was unattainable. The first prisoner does not know what the second prisoner intend to do, so he has to guard himself. The paradox lies in the fact that both prisoners has to accept a high penalty, in spite of a better solution for

² Games may be generalized to more agents with more choices, a n-persons game. In such games the influence from the single agent will be reduced with the size of the group. In this paper we will simulate repeated two person's games which enlarge the group of agents, and at least partly may be treated as a n-persons game (but still with two choices).

both of them. This paradox presumes that the prisoners were unable to talk to each other or take revenge after the years in jail. It is a symmetrical game with no background information.

In the original single play PD; two agents each have two options, to cooperate or to defect (not cooperate). If both cooperate, they receive a reward, R . The pay-off of R is larger than of the punishment, P , obtained if both defect, but smaller than the temptation, T , obtained by a defector against a cooperator. If the suckers payoff, S , where one cooperates and the other defects, is less than P there is a Prisoner's dilemma defined by $T > R > P > S$ and $2R > T+S$ (see Fig. 7.1). The second condition means that the value of the payoff, when shared in cooperation, must be greater than it is when shared by a cooperator and a defector. Because it pays more to defect, no matter how the opponent choose to act, an agent is bound to defect, if the agents are not deriving advantage from repeating the game. More generally, there will be an optimal strategy in the single play PD (playing defect). This should be contrasted to the repeated or iterated Prisoner's dilemma where the agents are supposed to cooperate instead. We will further discuss iterated games in the following sections.

The original Chicken game (CG), according to Russell [31] was described as a car race: "It is played by choosing a long straight road with a white line down the middle and starting two very fast cars towards each other from opposite ends. Each car is expected to keep the wheels of one side of the white line. As they approach each other, mutual destruction becomes more and more imminent. If one of them swerves from the white line before the other, the other, as he passes, shouts Chicken! and the one who has swerved becomes an object of contempt..."³

The big difference compared to Prisoner's dilemma is the increased costs for playing mutually defect. The car drivers should not really risk crashing into the other car (or falling off the cliff). In a chicken game the pay-off of S is bigger than of P , that is $T > R > S > P$. Under the same conditions as in the Prisoner's dilemma defectors will not be optimal winners when playing the chicken game. Instead there will be a combination between playing defect and playing cooperate, winning the game. In Fig. 7.1b R and P are assumed

³ An even earlier version of the chicken game came from the 1955 movie "Rebel Without a Cause" with James Dean. Two cars are simultaneously driving off the edge of a cliff, with the car driving teenagers jumping out at the last possible moment. The boy who jumps out first is "chicken" and loses.

to be fixed to 1 and 0 respectively. This can be done through a two steps reduction where in the first step all variables are subtracted by P and in the second step divided by R-P. This makes it possible to describe the games with only two parameters S' and T' (see Fig. 7.7 in the simulation section of this chapter). In fact we can capture all possible 2 x 2 games in a two-dimensional plane⁴.

a.	Cooperate	Defect
Cooperate	R	S
Defect	T	P

b.	Cooperate	Defect
Cooperate	1	$(S-P)/(R-P)$
Defect	$(T-P)/(R-P)$	0

Fig. 7.1 Pay-off matrices for 2 x 2 games where R = reward, S= sucker, T= temptation and P= punishment. In b the four variables R, S, T and P are reduced to two variables $S' = (S-P)/(T-P)$ and $T' = (T-P)/(R-P)$

As can be seen in Fig. 7.2 these normalized games are limited below the line $S' = 1$ and above the line $T' = 1$. CG has an open area restricted by $0 < S' < 1$ and $T' > 1$ whereas PD is restricted by $T' + S' < 2$, $S' < 0$ and $T' > 1$. If $T' + S' > 2$ is allowed there will be no upper limit for the value of the temptation. There is no definite reason for excluding this possibility (see also [12]). This was already pointed out when the restriction was introduced. “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’.” [28]. In this study no strategy explicitly make use of unilateral defections, so the extended area of PD is used.

⁴ Although there are an infinitely number of different possible games, we may reduce this number by regarding the preference orderings of the payoffs. Each agent has 24 (4!) strict preference orderings of the payoffs between it's four choices. This makes 24*24 different pairs of preference orderings, but not all of them represent distinct games. It is possible to interchange rows, columns and agents to obtain equal games. If all doublets are put away we still have 78 games left [29]. Most of these games are trivial because there is one agent with a dominating strategy winning.

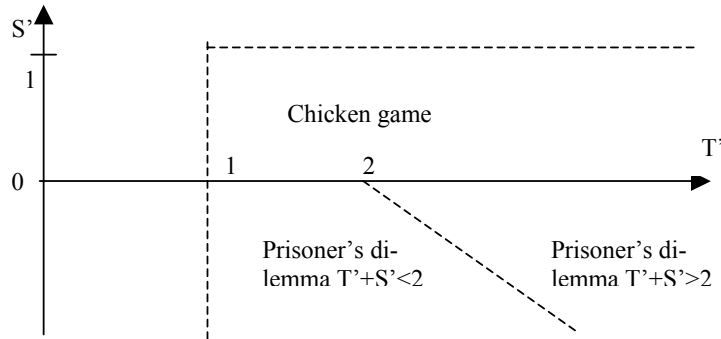


Fig. 7.2 The areas covered by Prisoner's dilemma and chicken game in a two-dimensional plane.

It is no coincidence that researchers have paid most interest to the Prisoner's dilemma and chicken games areas of the two-dimensional space. If we look at the left part ($T' < 1$) there will be no temptation, playing defect. If $S' > 1$ there will be no penalty for playing cooperate against playing defect.⁵

7.2.2 Evolutionary and iterated games

In evolutionary game theory ([25], [26]), 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 function being involved i.e. there is no way to anticipate the future by looking back into the memory. In many MAS, however, agents frequently use knowledge about other agents. There are at least three different ways of describing ESS from both an evolutionary and a MAS point of view.

Firstly, we define the ESS as a Nash equilibrium of different strategies. A Nash equilibrium describes a set of strategies where no agent unilaterally intend to change its choice. In MAS however, some knowledge about the other agents may be accessible when simulating the outcome of strategies.

⁵ Of course there are other interesting 2×2 plays, but this is outside the scope of this article. For an overview see [29]

Assume that agents can predict the behavior of their opponents from their past observations of play in “similar games”, either with their current opponents or with “similar” ones. If agents observe their opponents’ strategies and receive a number of observations, then each agent’s expectations about the play of his opponents converges to the probability distribution corresponding to the sample average of play he has observed in the past. 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. 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., [16]; [23]).

A game can be modeled as a *strategic* or an *extensive* game. A strategic game is a model of a situation in which each agent chooses its strategy once and for all, and all agents’ decisions are made simultaneously while an extensive game specifies the possible orders of events. An agent playing a strategic game is not informed of the plan of action chosen by any other agent while an extensive agent can reconsider its plan of action whenever a decision has to be made. All the agents in this chapter are playing strategic games.

According to the second way of describing the ESS, it can be described as a collection of successful strategies, given a population of different strategies. An ESS is a strategy (or possibly a set of strategies) such that if all the members of a population adopt it, then no mutant strategy (a strategy not in the current set of strategies) could *invade* (become a resident part of successful strategies) 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. The problem is that this is not the same as finding a successful strategy in an iterated game because in such games the agents are supposed to know the history of the moves. For non-trivial MAS and evolutionary systems, it is impossible to create a complete set of strategies. 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 be an ESS. If the given collection of strategies is allowed to compete in a population tournament, we will possibly find a winner, but not necessarily the same one for every repetition of the game. A population tournament allows successful strategies to be more common in the population of strategies when a new generation is introduced. In the simulation part of this

chapter we show some major differences between PD and CG in population tournaments.

Thirdly, the ESS can be seen as a collection of genetically evolving successful strategies i.e. combining a population tournament with the ability of introducing new generation strategies. It is possible to simulate a game through such a process, consisting of two crucial steps: mutation (i.e., a variation of the ways agents act) and selection (the choice of the preferred strategies). Different kinds of genetic computations (see, e.g., [18]: [17]: [20]) have been applied within the MAS society, but it is important to remember that the similarities to natural selection are restricted.⁶ For PD and CG mutational changes may occur by allowing strategies to change a single move (cooperate or defect) and then be subject to population selection. This method is not further expounded in this chapter.

7.2.3 Simulating iterated games

In an iterated game, unlike the repeated evolutionary game, the strategies are assumed to have a memory function. Most studies today look at the iterated Prisoner's dilemma (IPD) as a cooperative game where "nice" and "forgiving" strategies, like the *Tit-for-Tat* (*TfT*), are successful ([3]; [5]). A nice strategy is one, which never chooses to defect before the other agent defects, and a forgiving strategy does not retaliate a defect by playing defect forever. *TfT* simply follows the move of its opponent drawn in the round before. In iterated chicken game (ICG), mutual cooperation is less clearly the best outcome [22] but the situation is complicated. A mixed⁷ strategy may favor mutual cooperation.

Axelrod and Hamilton [5] introduced the concept of reciprocal altruism to game theory in their famous article "The evolution of cooperation". People were invited to submit their favorite strategy to an iterated Prisoner's dilemma game tournament [1]. The tournament was conducted as a round robin tournament where everyone met each other two by two. The only

⁶ Firstly, genetic algorithms use a fitness function instead of using dominating and recessive genes in the chromosomes. Secondly, there is a crossover between parents instead of the biological meiotic crossover.

⁷ With pure and mixed strategies we here refer to the set of strategies (played by individuals) winning the population game. A mixed strategy is a combination of two or more strategies from the given set of strategies i.e. an extended strategy set could include the former mixed strategy as a pure strategy.

known strategy for the participators in the beginning was the strategy *random*. For the tournament the *TfT* strategy was most successful, on average beating every other strategy. *TfT* starts with playing cooperatively and then repeats every move done by its antagonist. Axelrod [2] informed the participators about the result and invited them to a new extended tournament. Once again *TfT* was the major winning strategy.

Axelrod also conducted a population tournament where each strategy was allowed to survive into new generations of strategies. The proportion of the strategies depended on how successful each strategy was in the previous generation. In the end of the simulation there was typically only one successful strategy left. Again, *TfT* won most of the plays proving to be a robust strategy against these strategies.

The conclusions drawn by Axelrod were that nice, forgiving strategies like *TfT* defeat strategies playing defectively using threats and punishments. This is a remarkable conclusion because of:

In the single play PD playing defect is the winning strategy.

In both single play and repeated PD a defecting strategy always wins against a cooperating strategy.

TfT uses the advantage of being nice and forgiving when it meets itself. A defecting strategy always wins (or play even) against a nice strategy, but gets a low score when meeting other defecting strategies. A nice strategy will get a high score when meeting other nice strategies. This will compensate for the deficit in score against a defecting strategy. If there are a lot of cooperating strategies they will eliminate the defecting strategies of the competition.

Another complication that can be introduced to iterated games is the presence of noise. If there is uncertainty about the outcome, *TfT* will be less successful. In Axelrod's simulation, *TfT* still won the tournament when 1 per cent chance of misperception was added [3]. In other simulations of noisy environments, *TfT* has instead performed poorly [7]. The uncertainty represented by the noise reduces the payoff of *TfT* when it plays itself in the IPD.

Instead playing defect or playing a modified *TfT* strategy like *contrite-TfT* (*cTfT*, [10]), *Pavlov* [16] or *generous-TfT* (*gTfT*, [27]) may be more successful. *cTfT* (also called *Fair* [21]) is a modified version of *TfT* where the strategy is allowed to "apologize" or "be angry" instead of just repeating the opponent's move. *Pavlov* or *Simpleton* [3] cooperates if and only if the two competitors used the same move in the previous round. *gTfT* always

cooperate if the other agent cooperated in the previous round, but defect with a probability less than one if the other agent defected.

Ever since Axelrod presented his results there has been a lively discussion about the PD. As an example Axelrod is generally using the same payoff matrix for different simulations. Binmore [8] gives 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 *TfT* is not evolutionary stable. On the other hand evolutionary pressures will tend to select equilibrium for the IPD and ICG in which the agents cooperate in the long run.

7.2.4 Generous and greedy strategies

The principle for the categorization of strategies into nice and forgiving against defecting strategies, which uses threats and punishments, is unclear. Why is *TfT* not just treated as a strategy repeating the action of the other strategy instead?

One alternative way of categorize strategies is to group them together as being generous, even-matched, or greedy ([13]; [14]). If a strategy more often plays as a sucker, n_S , than playing temptation, n_T , then it is a generous strategy ($n_S > n_T$). An even-matched strategy has $n_S \approx n_T$ and a greedy strategy has $n_S < n_T$. n_S and n_T are the number of time an agent plays sucker and temptation respectively. Boerlijst, et al [9] uses a similar categorization into good or bad standings. A agent is in good standing if it has cooperated in the previous round or if it has defected while provoked, i.e., if the agent is in good standing it should not be greedy unless the other agent was greedy the round before. In every other case of defection the agent is in bad standing i.e. it tries to be greedy. The generous and greedy categorization uses a stable approach, a once and for all categorization, contrary to a more dynamic good and bad standing dealing with what happened in the previous move.

The stable approach of the generous and greedy categorization makes it easier to analyze this model. 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 n_S must equal the sum of the strategies n_T . 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 per cent co-operate $n_R + n_S$ when meeting another strategy. *AllC* will never act as a greedy strategy.

Always Defect (AllD) has 100 percent defect $n_T + n_P$ 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 $n_S \approx n_T$.

Random plays cooperate and defect approximately half of the time each. The proportions of n_S and n_T will be determined by the surrounding strategies.

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.

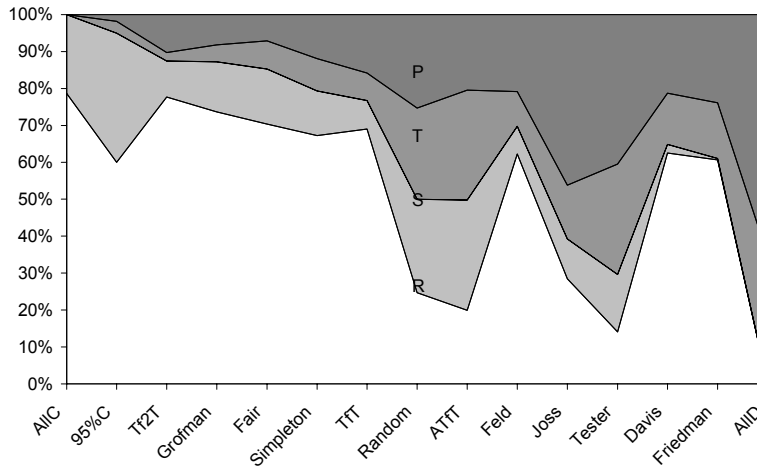


Fig 7.3: Proportions out of 100% of R, S, T, and P for different strategies

In the next section we use a simulation tool with 15 different strategies (see Table 7.1). We interpret the proportions of these strategies in Fig. 7.3 as a kind of context dependent fingerprint for the strategy in the given environment, independent of the actual value of the payoff matrix.

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 co-operation moves than *Random*, i.e. having significantly more than 50 per cent R or S. 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. The described relation between strategies is independent of what kind of game is played, but the actual outcome of the game is related to the payoff matrix.

7.3 The simulations

As mentioned earlier in this chapter the repeated Prisoner's dilemmas is regarded as cooperate games, i.e. games favoring agents playing cooperate. A typical winning strategy, like *TfT*, ends up in agents playing cooperate all the time. In chicken games the advantage of cooperation should be even stronger, because it costs more to defect compared to the Prisoner's dilemmas. Surprisingly, this is not the case when trying to analyze the chicken games. In the hawk and dove game [26], consisting of one PD and one CG part, the outcome for the CG is supposed to be a combination of playing cooperate and playing defect. We think this new "dilemma" can be explained by a larger robustness for the chicken games. This robustness may be present if more strategies are allowed and/or noise is introduced. In this chapter three different simulations comparing IPD and ICG are presented trying to verify this hypothesis.

Variants of Axelrod's original matrix—the first simulation used Axelrod's original payoff matrix for 36 different strategies. To investigate the differences we used 11 different matrices gradually moving from PD to CG [12].

Adding noise—5 different variants of Axelrod's matrix were used for 15 different strategies. Different levels of noise were added [13].

Normalized matrices—In all 209 different matrices were used for 15 different memory-0 and memory-1 strategies [11].

7.3.1 Variants of Axelrod's original matrix

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 fig. 7.4 a for each move of the Prisoner's dilemma:

a		
	C ₂	D ₂
C ₁	3, 3	0, 5
D ₁	5, 0	1, 1

b		
	C ₂	D ₂
C ₁	3, 3	1, 5
D ₁	5, 1	0, 0

Fig. 7.4 Example payoff matrix Prisoner's dilemma (4 a) and the chicken game (4 b).

In our experiment we use the same total payoff sum for the matrices as Axelrod used and a simulation tool involving 36 different strategies [24]. However, we vary the two lowest payoffs (0 and 1) continuously so that they change order between PD and the CG matrix in fig. 7.4 b.

It is a round robin tournament between different strategies with a fixed length of 100 iterations. Each tournament was run five times. Besides the two matrices above we varied P and S ten steps between 1 and 0 respectively without changing the total payoff sum for the matrix. As an example, (0.4; 0.6) means that a cooperate agent gets 0.4 meeting a defect and defect gets 0.6 meeting another defect. The different strategies are described in Mathieu and Delahaye [24].

We have used three characterizations of the different strategies:

Initial move—if the initial move of the strategy was cooperative, defect or random.

Nice—If the strategy does not make the first defect in the game.

Static—If the strategy is fully or partly independent of other strategies or if the strategy is randomized.

7.3.2 Adding noise to PD and CG

We developed a simulation tool in which 15 different strategies competed. Most of the strategies are described in ([1], [2]). In table 7.1 all the strategies

are described. 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.

	C ₂	D ₂
C ₁	1.5	2
D ₁	1	1.5+q

Fig. 7.5. A cost matrix for the resource allocation matrices.

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 (Fig. 7.5):

$$E_{avg}(S) = 1.5p(C,C) + 2p(C,D) + 1p(D,C) + (1.5+q)p(D,D) \quad (7.1)$$

We ran a simulation with the values for 1.5+q equal to: 1.6; 1.9; 2.1; 2.4; 3.0, and then we introduced noise on four levels: 0.01, 0.1, 1 and 10 per cent. This means that the strategies changed to the opposite moves for this given percentage.

7.3.3 Normalized matrices

The normalized study includes two different sets of simulations. In the first set, the strategies compete in a round-robin tournament with the aim of just to determine the tendency of different strategies to play cooperates and defect. In the second set, the competitive abilities of strategies in iterated population tournaments were studied within the IPD and the ICG.

In the simulations of the IPD and the ICG two sets of strategies were used. We used the strategies in Fig. 7.6 represented by finite automata [21]. The play between two automata is a stochastic process where all finite memory strategies can be represented by increasingly complicated finite automata. Memory-0 strategies, like *AllC* and *AllD*, does not involve any

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

Table 7.1. Description of the different strategies.

memory capacity at all. If the strategy in use only has to look back at one draw, there is a memory-1 strategy (a choice between two circles dependant

of the other agent's move). All the strategies in Fig. 7.6 belong to memory-0 or memory-1 strategies.

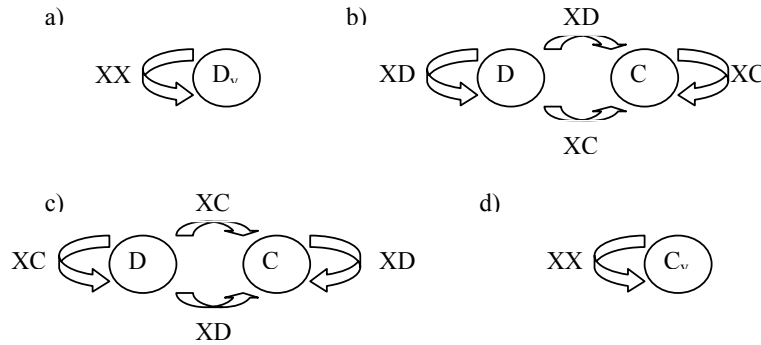


Fig. 7.6 a) *AllD* (and variants) b) *TfT* c) *ATfT* d) *AllC* (and variants).

On the transition edges, the left symbol correspond to an action done by a strategy against an pponent performing the right symbol, where an X denotes an arbitrary action. Y in C_y and D_y denotes a probability factor for playing C and D respectively.

Both sets of strategies include *AllD*, *AllC*, *TfT*, *ATfT* and *Random*. In the first set of strategies, the cooperative-set five *AllC* variants (100, 99.99, 99.9, 99 and 90 % probability of playing C) are added and in the second set of strategies, the defective-set the corresponding five *AllD* variants are added. C_y and D_y in Fig. 7.3 show a probability factor y 100, 99.99, 99.9, 99, 90 % or for the Random strategy 50% for playing C and D respectively.

	Cooperate (C)	Defect (D)
Cooperate (C)	1	$1-s_1$
Defect (D)	$1+s_2$	0

Fig. 7.7 A payoff matrix for PD and CG. C stands for cooperate, D for defect and s_1 and s_2 are cost variables. If $s_1 > 1$ it is a PD. If $s_1 < 1$ it is a CG.

To obtain a more general treatment of IPD and ICG, we used several variants of payoff matrices within these games, based on the general matrix of Fig. 7.7 (corresponding to Fig. 7.2).

In the first set of simulations we investigated the successfulness of the agents using different strategies (one strategy per agent) in a round-robin tournament. Since this is independent of the actual payoff value, the same round-robin tournament can be used for both IPD and ICG. Every agent was paired with all the other agents plus a copy of itself. Every meeting between agents in the tournament was repeated on average 100 times (randomly stopped) and played for 5000 times.

The result from the two-by-two meetings between agents using different strategies in the round-robin tournament was used in a population tournament. The tournament starts with a population of 100 agents for each strategy, making a total population of 900. The simulation halts when there is a *winning strategy* (all 900 agents use the same strategy) or when the number of generations exceeds 10.000. Agents are allowed to change strategy and the population size remains the same during the whole contest. For the IPD the following parameters were used: $s_1 \in \{1.1, 1.2 \dots 2.0\}$ and $s_2 \in \{0.1, 0.2 \dots 1.0, 2.0\}$, making a total of 110 different games⁸. For the ICG games with parameter settings $s_1 \in \{0.1, 0.2, \dots 0.9\}$ and $s_2 \in \{0.1, 0.2, \dots 1.0, 2.0\}$ a total of 99 different games were run. Each game is repeated during 100 plays and the average success is calculated for each strategy. For each kind of game there is both the cooperative-set and the defective-set.

7.4 Results

7.4.1 Variants of Axelrod's original matrix

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 agent 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

⁸ For the strategies used in this simulation the constraint $2R > T + S$ does not affect the results, so these combinations are not excluded.

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.

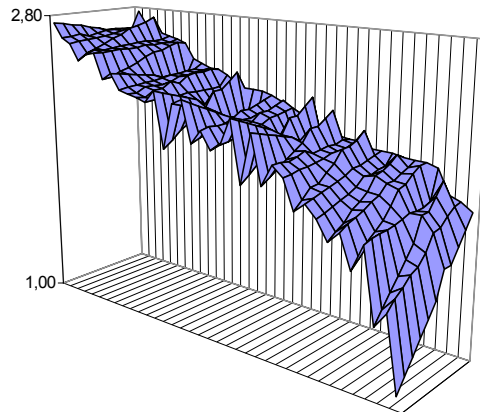


Fig.7.8. Comparing PD and CG. In the figure above the CG is in the foreground and the PD in the background, the best strategies are to the left and the worst to the right.

The top scoring strategies 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.

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

Our simulation indicates that the chicken game to a higher extent rewards cooperative strategies than the Prisoner's dilemma because of the increased cost of mutual defections. The following parts of the result confirm these statements:

All the top six strategies are nice and start with cooperation. They have small or moderate differences in scores between the chicken game and Prisoner's dilemma. *TfT* is a successful strategy but not the best. All the 11 strategies, with a lower score than *random*, either starts with defect or, if

they start with cooperation, is not nice. All of these strategies are doing significantly worse in the CG than in the PD. This means that we have a game that benefits cooperators better than the PD, namely the CG.

A few of the strategies got, despite of the overall decreasing average score, a better score in the CG than in the PD. 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. 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 strategies⁹. All static strategies, which cooperate on the first move, and some of the partially static ones, do better in the CG than in the PD. We interpret this result to be yet another indicator of the importance of being forgiving in a CG.

7.4.2 Adding noise to PD and CG

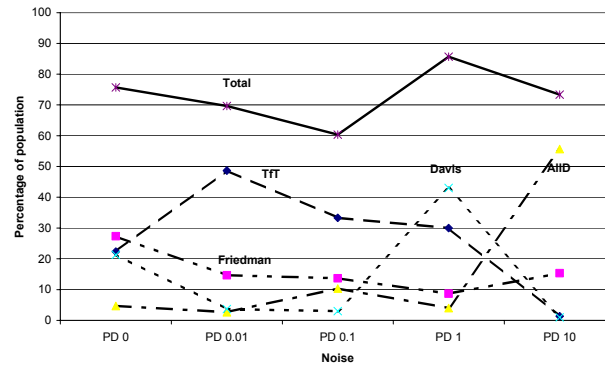


Fig. 7.9 The four most successful strategies in PD games with increasing noise. Total represents the percentage of the population these four strategies represented.

⁹ In 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) also would probably do better in a CG than in a PD, but such strategies were not part of our simulations.

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. 7.9 and Fig. 7.10 show these strategies for PD and CG when 0, 0.01, 0.1, 1.0, and 10.0 per cent noise is introduced. Among the four most successful strategies in PD there were three greedy and one even-matched strategy (Fig. 7.9. see also Fig. 7.3.). 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.

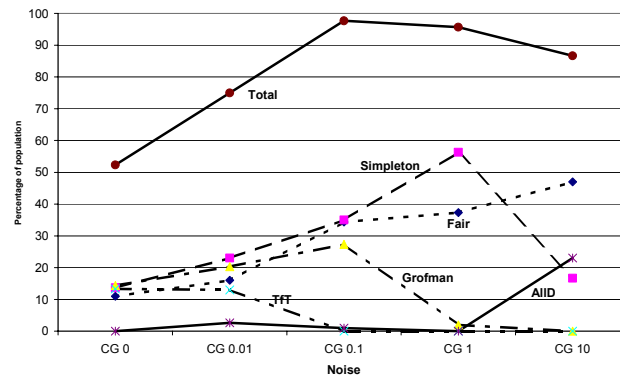


Fig. 7.10 The five most successful strategies in CG games with increasing noise. Total represents the percentage of the population these five strategies represented.

Three out of five of the most successful strategies in CG were generous. The total line in Fig. 7.10. 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. *Simpleton* that declined after 1% noise level showed the same pattern. Only *Fair* continued to increase when more noise was added, making it a dominating strategy at 10% noise together with the greedy strategy *AllD*.

7.4.3 Normalized matrices

7.4.3.1 Playing random

If agents with a number of random strategies are allowed to compete with each other, they will find a single winning strategy after a number of generations. This has to do with genetically drift and small simulation variations between different random strategies about how they actually play their C and D moves. As can be seen in Fig. 7.11 there are an increasing number of generations for finding a winning strategy when the total population size increases. This almost linear increase ($r = 0.99$) is only marginally dependent of what game is played.

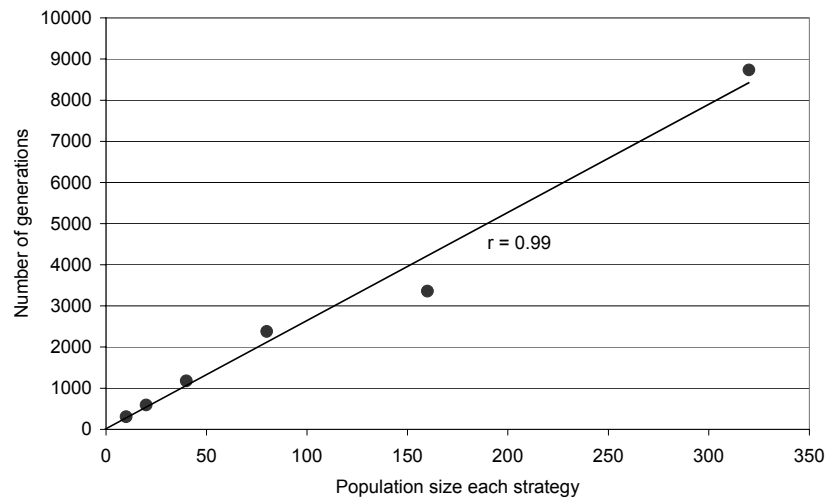


Fig. 7.11. Number of generations for finding a winning strategy among 15 random strategies with a varying population size

The simulation consists of strategies with a population size of 100 individuals each. Randomized strategies with 100 individuals are, according to Fig. 7.11., supposed to halt after approximately 2800 generations in a population game. There are two possible kinds of winning strategies: *pure strategies* that halt and *mixed strategies* (two or more pure strategies) that do

not halt. If there is an active choice of a pure strategy it should halt before 2800 generation, because otherwise playing random could be treated as a winning pure strategy. Fig. 7.12 shows the relations between pure and mixed strategies for IPD and ICG. For all 110 games each run with one cooperative-set and one defective-set within IPD this is true. For the ICG only one out of 99 different games halted before 2800 generations. This game ($T=1.1$, $S=0.1$) was very close to an IPD. For the rest of the ICG there was a mixed strategy outcome. There is no reason to believe us to find a single strategy solution by extending the simulation beyond 10000 generations. If there exists a pure solution, this solution should turn up much earlier.

7.4.3.2 Pure and mixed strategies for cooperative and defective sets.

Fig. 7.12 shows a major difference between pure and mixed-strategies for IPD and ICG. IPD has no successful mixed strategies at all, while ICG favors mixed-strategies for a overwhelming majority of the games. Some details not shown in Fig. 7.12 are discussed below.

	IPD		ICG	
	Cooperative-set	Defective-set	Cooperative-set	Defective-set
Pure strategies	<i>TfT</i> 78% <i>AllD</i> 20%	<i>TfT</i> 75% <i>AllD</i> 20%	<i>TfT</i> 3%	<i>TfT</i> 2%
Mixed strategies	none	none	2-strat. 61% 3-strat 33%	2-strat 69% 3-strat 24%

Fig. 7.12. The difference between pure and mixed-strategies in IPD and ICG. For details see text.

For the cooperative-set there is a single strategy winner after on average 167 generations. *TfT* wins 78% of the plays and is dominating 91 out of 110 games¹⁰. *AllD* is dominating the rest of the games and wins 20% of the plays.

For the defective-set there is a single strategy winning in 47 generations on average. *TfT* is dominating 84 games, *AllD* 21 games and *99.99D*, playing D 99.99% of the time, 5 games out of 110 games in all. *TfT* wins 75% of the plays, *AllD* 20% and *99.99D* 4%.

In the cooperative-set there are two formations of mixed-strategies winning most of the games; one with two strategies and the other with three

¹⁰ A game is dominated by a certain strategy if it wins more than 50 out of 100 plays.

strategies involved. This means that when the play was finished after 10000 generations not a single play could separate these strategies finding a single winner. The two-strategy set *ATfT* and *AllD* wins 61 % of the plays and the three-strategy set *ATfT*, *AllD* and *AllC_{tot}* wins 33% of the plays. *AllC_{tot}* means that one and just one of the strategies *AllC*, *99.99C*, *99.9C*, *99C* or *90C* is the winning strategy. For 3% of the games there was a single *TfT* winner within relatively few generations (on average 754 generations).

In the defective-set there is the same two formations winning most of the games. *ATfT* + *AllD_{tot}* wins 69% of the plays and *ATfT* + *AllC* + *AllD_{tot}* wins 24% of the plays. *AllD_{tot}* means that one and just one of the strategies *AllD*, *99.99D*, *99.9D*, *99D* or *90D* is the winning strategy. *TfT* is a single winning strategy in 2% of the plays, which needs on average 573 generations before winning a play.

7.4.3.3 Generous and greedy strategies in IPD and ICG

In the C-variant set all *AllC* variants are generous and *TfT* is even matched. *AllD*, *ATfT* and *Random* are all greedy strategies. In the D-variant set all *AllD* variants are greedy and *TfT* is still even-matched. *AllC*, *ATfT* and *Random* are now representing generous strategies.

In the IPD the even-matched *TfT* is a dominating strategy in both the C- and D-variant set with the greedy *AllD* as the only primary alternative. So the IPD will end up being a fully cooperative game (*TfT*) or a fully defecting game (*AllD*) after relatively few generations. This is the case both for the C-variant set and, within even fewer generations, for the D-variant set.

In ICG there is instead a mixed solution between two or three strategies. In the C-variant *ATfT* and *AllD* form a greedy two-strategy set¹¹. In the three-strategy variant the generous *AllC_{tot}* join the other two. In all, generous strategies only constitute about 10% of the mixed strategies. In the D-variant the generous *ATfT* forms various strategy sets with the greedy *AllD_{tot}*.

7.5 Discussion

In our first study of variants of Axelrod's original matrix a CG tends to favor cooperation more than a PD because of the values of the payoff matrix. The

¹¹ With just *ATfT* and *AllD* left *ATfT* will behave as a generous strategy even though it starts off as a greedy strategy in the C-variant environment.

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.

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 ([6]: [7]). 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.

In the normalized matrices stochastic memory-0 and memory-1 strategies are used. The main difference between IPD and ICG is best shown by the two strategies *TfT* and *ATfT*. *TfT* does the same as its opponent. This is a successful way of behaving if there is a pure-strategy solution because it forces the winning strategy to cooperate or defect, but not doing both. *ATfT* is doing very badly in IPD because it tries to jump between playing cooperate and defect.

In ICG we have a totally different assumption because a mixed-strategy solution is favored (at least in the present simulation). *ATfT* does the opposite as its opponent but cannot by itself form a mixed-strategy solution. It has to rely on other cooperative or defect strategies. In all different ICG *ATfT* is one of the remaining strategies, while *TfT* is only occasionally winning a play.

For a simple strategy setting like the cooperative and defective-set, ICG will not find a pure-strategy winner at all but a mixture between two or more strategies, while IPD quickly finds a single winner.

Unlike the single play PD, which always favors defect, the IPD will favor playing cooperate. In CG the advantage of cooperation should be even stronger, because it costs more to defect compared to the PD, but in our simulation greedier strategies were favored with memory-0 and memory-1 strategies. We think this new paradox can be explained by a larger “robustness” of the chicken game. This robustness may be present if more strategies, like the strategies in the two other simulations, are allowed and/or noise is introduced. Robustness is expressed by two or more strategies winning the game instead of a single winner or by a more sophisticated single winner. Such a winner could be *cTfT*, *Pavlov*, or *Fair* in the presence of noise, instead of *TfT*.

In Carlsson and Jönsson [14] 15 different strategies were run in a population game within different IPD and ICG and with different levels of noise. *TfT* and greedy strategies like *AllD* dominated the IPD while *Pavlov* and two variants of *cTfT* dominated the ICG. For all levels of noise it took on average fewer generations to find a winner in the IPD. This winner was greedier than the winner in the ICG. If instead a lot of non-intuitive strategies were used together with *AllD*, *AllC*, *TfT* and *ATfT*, IPD very quickly terminated with *TfT* and *AllD* winning the games, while ICG did not terminate at all for most of the different noises.

We propose that the difference between IPD and ICG can be explained by pure and mixed-strategy solutions for simple memory-0 or memory-1 strategies. For simple strategies like *TfT* and *ATfT*, ICG will not have a pure-strategy winner at all but a mixture between two or more strategies, while IPD quickly finds a single winner. For an extended set of strategies and/or when noise is present the ICG may have more robust winners than the IPD by favoring more complex and generous strategies. Instead of *TfT* a complex strategy like *Fair* is favored.

From an agent engineering perspective the strategies presented in this chapter are quite simple. The presupposed agents are modeled in a predestined game theoretical environment without a sophisticatedly internal representation. If we give the involved agents the ability to establish trust the difference between the two kinds of games are easier to understand. In the PD establishing trustworthiness between the agents means establishing trust, whereas in CG, it involves creating fear, i.e. avoiding situations where there are too much to lose. This makes CG a strong candidate for being a major cooperate game together with PD.

Acknowledgements

The author wishes to thank Paul Davidsson, Stefan Johansson, Ingemar Jönsson and the anonymous reviewers from the IAT conference for their critical reviews of previous version of the manuscript and Stefan Johansson for running the simulation.

Bibliography

- [1] Axelrod, R. Effective Choice in the Prisoner's Dilemma *Journal of Conflict Resolution* vol. 24 No. 1, p. 379-403, 1980a.
- [2] Axelrod, R. More Effective Choice in the Prisoner's Dilemma *Journal of Conflict Resolution* vol. 24 No. 3, p. 3-25, 1980b.
- [3] Axelrod, R., The Evolution of Cooperation. Basic Books, New York 1984.
- [4] Axelrod, R., and Dion, D., The further evolution of cooperation *Nature*, 242:1385-1390, 1988.
- [5] Axelrod, R., and Hamilton, W.D., The evolution of cooperation. *Science* 211, 1390, 1981.
- [6] Bendor, J., and Kramer, R.M., and Stout S., "When in Doubt... Cooperation in a Noisy Prisoner's Dilemma." *Journal of conflict resolution* vol. 35 No 4 p. 691-719, 1991.
- [7] Bendor, J., "Uncertainty and the Evolution of Cooperation" *Journal of Conflict resolution* vol. 37 No 4 p. 709-734, 1993
- [8] Binmore, K. Playing fair: game theory and the social contract The MIT Press Cambridge, MA, 1994.
- [9] Boerlijst, M.C., Nowak, M.A. and Sigmund, K., Equal Pay for all Prisoners. / The Logic of Contrition *ILASA Interim Report IR-97-73* 1997.
- [10] Boyd, R., Mistakes Allow Evolutionary Stability in the Repeated Prisoner's Dilemma Game, *J. Theor. Biol.* 136, pp. 47-56, 1989.
- [11] Carlsson, B., How to Cooperate in Iterated Chicken Game and Iterated Prisoner's Dilemma *Intelligent Agent Technology* pp. 94-98 1999.
- [12] Carlsson, B. and Johansson, S. "An Iterated Hawk-and-Dove Game." In *W. Wobcke, M. Pagnucco, C. Zhang ed. Agents and Multi-Agent Systems Lecture Notes in Artificial Intelligence* 1441 p. 179—192, Springer-Verlag, 1998.
- [13] Carlsson, B., Johansson, S. and Boman, M., Generous and Greedy Strategies *Proceedings of the Congress on Complex Systems*, Sydney 1998.
- [14] Carlsson, B. and Jönsson, K.I., The fate of generous and greedy strategies in the iterated Prisoner's Dilemma and the Chicken Game under noisy conditions. Manuscript 2000.
- [15] Durfee, E.H., Practically Coordinating *AI Magazine* 20 (1) pp. 99-116, 1999.
- [16] Fudenberg, D. and Maskin, E., Evolution and cooperation in noisy repeated games, *American Economic Review* 80 pp. 274-279, 1990.

- [17] Goldberg, D. Genetic Algorithms Addison-Wesley, Reading, MA, 1989.
- [18] Holland, J.H. Adaptation in natural and artificial systems MIT Press, Cambridge, MA, 1975.
- [19] Jennings, N. and Wooldridge, M., "Applying Agent Technology" in *Applied Artificial Intelligence*, vol.9 No.4 p 357-369 1995.
- [20] Koza, J. R. Genetic Programming On the Programming of Computers by Means of Natural Selection. The MIT press, Cambridge, MA, 1992.
- [21] Lindgren, K., Evolutionary Dynamics in Game-Theoretic Models in *The Economy as an Evolving Complex System II (Arthur, Durlauf and Lane eds. Santa Fe Institute Studies in the Sciences of Complexity, Vol XXVII) Addison-Wesley*, 1997.
- [22] Lipman, B.L., Cooperation among egoists in Prisoner's Dilemma and Chicken Game. *Public Choice* 51, pp. 315-331, 1986.
- [23] Lomborg, B., Game theory vs. Multiple Agents: The Iterated Prisoner's Dilemma, in *Artificial Social Systems (C. Castelfranchi and E. Werner eds. Lecture Notes in Artificial Intelligence 830)* 1994.
- [24] Mathieu, P., and Delahaye, J.P., <http://www.lifl.fr/~mathieu/ipd/>
- [25] Maynard Smith, J. and Price, G.R., The logic of animal conflict, *Nature* vol. 246, 1973.
- [26] Maynard Smith, J., Evolution and the theory of games, Cambridge University Press, Cambridge 1982.
- [27] Molander, P., The optimal level of generosity in a selfish, uncertain environment, *J. Conflict resolution* 29 pp. 611-618 1985.
- [28] Rapoport, A. and Chammah, A.M., Prisoner's Dilemma A Study in Conflict and Cooperation Ann Arbor, The University of Michigan Press 1965.
- [29] Rapoport, A. and Guyer, M. *A taxonomy of 2 x 2 games Yearbook of the Society for General Systems Research*, XI:203-214, 1966.
- [30] Rosenschein, J. and Zlotkin, G., Rules of Encounter, MIT Press, Cambridge, MA, 1994.
- [31] Russell, B., Common Sense and Nuclear Warfare Simon & Schuster 1959.
- [32] Selten, R., Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game theory*, 4:25-55, 1975.