Via Po, 53 – 10124 Torino (Italy)
Tel. (+39) 011 6702704 - Fax (+39) 011 6702762
URL: http//www.de.unito.it

WORKING PAPER SERIES

BEATING THE TIT FOR TAT: USING A GENETIC ALGORITHM TO BUILD AN EFFECTIVE ADAPTIVE BEHAVIOR

Gianluigi Ferraris e Magda Fontana

Dipartimento di Economia "S. Cognetti de Martiis"

Working paper No. 04/2006

Università di Torino

Beating the Tit for Tat: Using a Genetic Algorithm to Build an Effective Adaptive Behavior

Gianluigi Ferraris
University of Torino – Ferraris@econ.unito.it

Magda Fontana
Department of Economics "S. Cognetti de Martiis"
Via Po 53, 10124 Turin – Italy
magda.fontana@unito.it

Abstract: Agents capable of adaptive behavior can be obtained by means of AI tools. Thanks to these, they develop the ability to vary their Behavior in order to achieve satisfying results in the simulated environment. In the paper, artificially intelligent agents play an iterated prisoner's dilemma against agents that reproduce (in a fix way) strategies that have emerged in Axelrod's tournament. The objective of the adaptive agent is to earn a payoff higher than one of the Tit-for-tat, the strategy which has shown the better performance in the Axelrod's experimental setup. In the work, Genetic Algorithms are employed to produce and modify rules that are apt to achieve the set task. The adaptive dynamics is analysed in depth in order to understand the issues related to the codification of knowledge and to the evaluation of diverse strategies. In order to highlight different nuances of these matters we have amended the method as to improve it and experimented different knowledge's codifications.

Introduction

In his tournaments, Axelrod [1] found out that among the numerous submitted strategies the best performing was the tit-for-tat (TfT), the strategy that implies to respond to the opponent by replicating her last strategy. Thus, the TfT seems to be the best available option in a world of agents with pre-defined, non-adaptive, non-selective behaviour. In fact, Axelrod's agents reacted to the opponents' strategies by simply recalling previous interactions. This paper aims at demonstrating that an adaptive behaviour can give better results that a predetermined one. An adaptive agent can build selective rules to be adopted with single agents or with categories of players even in the absence of a previous description of the behaviour of its opponents. To this purpose, the used adaptive agents have been endowed with the ability of: i) modifying their rules ii) mastering rules for the interaction with specific opponents. AI methods [2] can be a helpful tool to enrich the modelling of adaptive behaviour; however, their proper use is not trivial. For instance, the imprecise representation of the rules, mistakes in passing information or in computing the fitness of a strategy can reduce the effectiveness of the methods and the goodness of results.

Among the AI methods, the so-called evolutionary algorithms (Genetic Algorithms, Classifier Systems and Genetic Programming) seem to be more suited for our research since they produce easily interpretable structures to the advantage of the analysis of their adaptation and do not require a preliminary training and therefore reduce the external influences of the researcher. In the present work we lean on GA, however further development will include CS and the comparison with the Neural Network to test the contribution of the connessionistic approach.

As a first step, the criteria of efficiency and effectiveness used to appreciate the results obtained by the agents have been set forth. As for effectiveness, the main criterion is the ability to over perform the TfT. The remaining criteria are described in the following pages. An experimental setup has then been devised. This is based on the ABM paradigm [3] used for the study *in silico* of complex emergent phenomena. The code has been realised by using the SWARM's libraries. To allow for future extensions, the framework has been generalised in order to encompass different games. The model includes five types of agents with pre-determined rules.

After validation, conducted by observing each kind of non-adaptive agent in isolation, AI agents have been included. Different experimental scenarios have been single out and, for each of them, experiments have been repeated by using different seeds for the generation of random numbers. The systematic success of the adaptive agent let us to state with reasonable robustness that it can beat the TfT. The observation of the evolutionary dynamics of the rules has permitted to study their adaptation and the factors that have determined their success. This issues, together with some suggestions for the use of the GA, lie at the core of the present work.

ABM and Swarm

The Agent-based modelling paradigm [3] focuses on role of the agent as an autonomous entity, which is able to generate strategies. Aggregate patterns emerge in the simulation because of the local interaction among entities. ABM has become one of the most acknowledged tool for the study of complexity in its evolutionary features. In our simulations, the agent is identified with the single player.

The model has been written in SWARM, a free tool, which has been realised about ten years ago at the SFI [4] and continuously updated by a wide community of users. The use of Swarm allows for an easy sharing of findings of the experiments and of the experiment itself. In addition, it provides a set of function for the observation of the results. The typical Swarm structure is articulated in three levels: the Observer, that includes the programs that record and publish the data during the run of the simulation; the Model, that creates and governs the objects that comprise the model; the Agent which is the representation of the elements that constitute the model's population. The Model creates the Agents and sends them instructions relative to which function to activate and its timing (scheduling). Results can be published in different graphical forms thanks to the versatility of the Observer. The structure is hierarchical: the Observer creates the Model while the latter creates the agents.

The architectural scheme of the model, ERA (Environment Rules Agents), has been borrowed by Terna [5]. ERA divides the models in four layers each with specific functions and then disciplines communication among them and within each layer. In ERA, there is a separation among: i) environment, ii) agents, iii) rule masters and iv) rule makers. The agent is distinct from its set of rules and its behaviour derives from the hints provided by the rule master on the ground of what decided by the programmer or by the rule maker. The adoption of such a scheme leads to clearer architectures in order to improve the possibility to share the researches and to make it easier to amend the program.

Artificial Intelligence, evolutionary methods, and genetic algorithms

The term artificial intelligence is used to define the discipline, the theories, and the methods to build artificial systems that constitute computational models of cognitive processes, and that result in behaviors that we would be inclined to call intelligent. According to the relative importance given to each of the two aspects we can distinguish between a “strong” and a “weak” paradigm of artificial intelligence. The former refers to the reproduction of the functioning of the human intelligence, while the latter sticks to a sensible behaviour of the machine. In the present work, we adhere to the weak paradigm.

The basic hypothesis in the AI research is known as “the hypothesis of the physical symbol system”. It is partially overcome by the sub-symbolic approach of the neural networks and genetic algorithms but still has a good explanatory power if one interprets the symbol as containing only bits of information. The formulation of the hypothesis and the definition of the physical symbol can be found in Newell e Simon [6].

From the hypothesis of the symbols, it derives the [7] need to have knowledge in order to obtain an intelligent behavior. Thanks to the symbol, knowledge can be represented, memorised and manipulated to obtain new structures. That is to say, new knowledge that increases the one already owned. The observation of the process of natural evolution, in the light of the DNA discovery, has lead to interpret the proteins that it contains as conveying all the knowledge necessary for the development of a very complex object: the phenotype (a living being). In the AI language, we could say that the symbol system used to codify knowledge embedded in the genotype is made up by four elements only. Genes could be intended as rules, highly complex bits that are able to contain a large amount of knowledge. Given the bounded extension of the symbol system, the meaning conveyed by each of the four bases is a very small portion of the knowledge carried by the gene. The latter derives by the different combinations of the four bases. This has two benefits: the risk of completely losing information is strongly reduced and the operations necessary to act on information are extremely simplified.

The AI techniques that exploit a codification of knowledge and mechanisms for its modification and adaptation which are similar to the natural ones are known as “evolutionary”. In this category falls the Genetic Algorithms and the Classifier Systems.

Their functioning recalls the Darwinian process of natural selection: the best individuals endure and spread while the less fit are bound to extinction. Sexual reproduction is obtained through copying and crossing the parents. During the process of copying, there is the possibility for genetic mutation: a mistake can change the information contained in a single gene. In artificial evolution, the evaluation of individuals is a positive function of the outcome of their action in the surrounding environment.

Goldberg [8] defines genetic algorithms as search algorithms based on natural selection and genetics. They represent the principle of the survival of the fittest by means of a mechanism of information exchange, structured but stochastic, in such a way that the search algorithm exhibits some of the innovative nose typical of the human search. In each generation, a new set of artificial creatures (sequences) is generated on the ground of the better individuals of the previous set.

When using GA [9], the search for the solution to a program is characterised as the attempt to express a structure which could represent actions. These, in turn, must be able to lead the system to interact with the environment in an increasingly fit way. In practice, the objective is to determine which organism is better adapted to a given context. When environmental constraints are not known or are emergent the analysis of the evolution of the rules allows to draw inferences on the rules of functioning of the system under study.

The strength of the method derives from its ability to adapt and therefore to obtain ever-increasing structures. Expressions are codified in binary strings. These are formed by lining up 1 and 0 up to the point in which all the knowledge necessary for the posed objective has been codified. Each string is an individual and their set represents the population subject to evolution. Crucial to the understanding of the functioning of the method is the concept of schema [9] which is a part of a string. Individuals with the highest evaluation will probably carry schema which are better than those carried by individuals with a lower fitness. It is for this reason that individuals with a high fitness are selected for reproduction while individuals with a low evaluation are sent to extinction. Reproduction mixes the schema therefore widening the span for the generation of novel structures.

GA works exactly according to this process by asking to the user to evaluate each individual before each evolution cycle. Evolution is simulated by selecting a given fraction of the population for reproduction and another for extinction. The choice follows a stochastic process so that a fit individual has a good chance of being selected for reproduction but it is not certain that it will be. On the same ground, an individual with low fitness can be selected for evolution. Reproduction takes place by reproducing the parents and then by switching the string at a randomly established point. A further manipulation aimed at widening the set of solutions occurs when (with a very low probability) the newborn structure is changed by arbitrary mutation in one of the symbol of the strings. The probability of extinction is an inverse function of the fitness. Iteration of the process leads the population to converge on a single type the fittest among those experienced. It is worth noting that there in no guarantee that the selected individual will be the optimal one. One must therefore be wary in order to avoid premature convergence paying attention to the fitness function and fitness evaluation. In particular, the structure must receive a homogeneous evaluation in order not to impair the adaptation process.

The game

The interaction between adaptive agents and agent with fixed beahviour recalls the well-known Axelrod's IPD [1]. Players meet in pairs and earn a payoff that depends on the opponent's reply to the proposed strategy. The following bi-matrix shows the payoffs:

Table 1	*Cooperate*	*Defect*
Cooperate	3,3	0,5
Defect	5,0	1,1

Efficiency and Effectiveness criteria

The use of intelligent systems as a tool for scientific research renders crucial the judgement relative to the performance of the adopted methods. Traditionally, due to its industrial use, the emphasis has regarded mainly computational efficiency and the ability to learn while the scientific use of GA requires a more fine-grained series of criteria. In our particular setting, one must be acquainted with the features of the adaptive process in order to support our statement that an adaptive agent can beat the one with fixed rules even in a very simple situation with a known outcome. Moreover, the decision to use of strongly autonomous agents must be supported with an adequate and precise knowledge of how this autonomy can affect the simulation. Results must be analysed critically, being focused on the issue of reliability and sensitivity, on the methodological choice and on the set of parameters.

Having stated our goal, that is beating the TfT, we suggest three orders of evaluation criteria: i) effectiveness, ii) efficiency, iii) simplicity in use.

Effectiveness will be judged according to:

1. Stability of strategies.
2. Memory span, that is the ability to operate with a high number of strategies.
3. Segregation of strategies, that is the ability to isolate strategies that are not currently applied. The ability to create such kind of speciation implies that an algorithm can satisfactorily face a given number of different events.
4. Rielaboration, of unfit rules, that is the ability to quickly modify rules that become unfit due to changes occurred in the environment.
5. Rapidity in recalling strategies that have not been used for a long time due to their ineffectiveness.
6. Graceful degradation: ability to cope with imperfect information.

For what concerns efficiency we will consider the number of cycles of interaction with the environment, necessary to reach the optimal or satisfying result.

Ease in use will be judged according to the friendliness in the representation of strategies, in input codification and in the transparency of interpretation of the produced rules.

The model

According to the "mental" interpretation of evolutionary algorithms [10], the structures produced by the GA are assimilated to the ideas that populate the mind of agents. Rules are evaluated through the application of the phenotype as derived by the agent's interpretation of the structures (genotype). Exchange of ideas between agents can take place only by explicit communication. This interpretation seems more plausible than its opposite, the individual one [10] in that it does not allow the intersection of structures pertaining to different rule makers. In evolutionary terms, this could happen through reproduction but it would nonetheless imply that an individual could read another individual's mind.

Simulations consist in running a given number of round robin tournament in which each agent is paired with the rest of the population. An object called Club deals with the scheduling of actions asking the agents to play in a given order which is different for every turn. The list of agent is shuffled before each tournament so that the order of pairing continuously changes. Once the agent receives the order to act, he must collect the necessary information (i.e. the type of opponent). He then asks to its rule master to provide the behaviour to follow. The rule master is different for entities operating according to determined rules and agents that use a GA. Based on what suggested by the rule maker the agent act by sending its strategy to the Club. This records the strategy and asks to the object called Game to compute the payoff. The amount of the earned payoff is passed to the agent by the Club, and in the case of adaptive agent, to the GA as an evaluation of the fitness.

The model uses five types of agents with fixed rules:

i) Perpetual cooperator: which always cooperates;

ii) Perpetual defector: which always defects;

iii) Noise player: which randomly picks a strategy from a uniform distribution;

iv) Tit-for-tat (hereafter TfT): This starts with a cooperation and than always plays by reproducing the strategy encountered in the last interaction.

v) Tit-for-2tat (hereafter TF2T): is a more tolerant player than TfT. It forgives the first defection before turning itself into a defector.

In addition, we can generate instances of agents operating on the ground of rules elaborated by a GA. It is worth noting that when there is only one instance for each kind of agent, the information regarding the type of agent will coincide with its identity, while when there is more than one agent per type this does not hold. It follows that the decision concerning the number of agents alters considerably the information available to the agent. In the first case, the player knows the identity of the opponent; in the second case, he only knows its typology. Moreover, having more agents of the same type would permit intra-type encounters. This setting could be used to analyse reciprocal dependency that could result in the adaptation process.

In order to guarantee the independence of the agents, each of them has been attributed, when necessary, a distribution of pseudo-random numbers. An analogous procedure has been followed in the ModelSwarm, the Club, and the objects that genetically manipulate the rules. We have use uniform distribution and the plurality of them tend to guarantee their effectiveness even in the case of extraction of few numbers.

Accuracy in computation has been tested by running trial simulations in the presence of agents with fixed rules only. The correspondence of theoretical results, computed a priori, and those produced by the model has allowed validating it. In more details:

1. The agents with fixed rules have been introduced, one type at the time, and their actions and the earned payoff verified.
2. Controls have being repeated both in the round robin tournament and in random match of agents.
3. The statistical neutrality (for large number) of the introduction of the noise player has been tested.

The simulation

We have run simulation of the following scenarios:

1. A GA against a single instance of agents with fixed rules except for the Noise player.
2. A GA against a single instance of agents with fixed including the Noise player.
3. A GA against more than one agent of a given type except for the Noise player.
4. A GA against more than one agent of a given type including the Noise player.
5. A GA against a single instance of 'mutant' agents with fixed rules
6. A GA with an initial population with a majority of cooperative rules against a single instance of agents with fixed rules.

The first scenario, allows for analysis of the criteria of effectiveness sub 1 and 2. The second one tends to appreciate the ability of the GA to act under imperfect information (criterion sub 6). In the third scenario, the change in the information concerns mainly the case for Tit for Two Tat. Toward that kind of agent, the optimal strategy is to alternate cooperation to defection in order to exploit its forbearance. This operates univocally when there is only one instance of the Tf2t, while when there are many, the alternation could be fallacious. Imagine having three Tf2t agents say a, b, c. If I have defected with a, my best option is to cooperate with a on the next run. However, the GA does not known the single agent but only its type, so it may happen that it will cooperate with b and defect with c with no guarantee of convergence on some optimal strategy. In this case, the achievement of good performance could demonstrate a significant robustness of the method in showing its ability to generalise its behaviour from an agent to a type of agents. The scenario 4 tends to appreciate the ability of the GA to preserve good strategies even when they are applied seldom (criterion sub 3). Setting 5 introduces the ability for agents with fixed rules to disguise their identity, that is to say that they can declare a type different from their actual one. For instance, a perpetual cooperator while sticking to its rules could define itself as a perpetual defector. Since the identification has proven to be the better representation of knowledge, this experiment aims at verifying its ability to adapt to changeable situations (criteria sub 4 and 5).

Since the optimal strategy implies a majority of defections, the fifth scenario aims at testing the ability of the GA to compute the adaptation of strategies starting from an unfit endowment of rules.

In addition to the single tests, the plurality of scenarios constitutes a good benchmark for the robustness of results to different parameters of the model.

Quantitative indexes used for the evaluation of effectiveness

The following matrix shows the maximum payoff that an adaptive agent can extract from a single instance of agent with predetermined rules except for the Noise player.

Table 2: optimal results

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*Total*
Perpetual cooperator		0	3	3	0	6
Perpetual defector	5		1	1	1	8
TfT	3	1		3	3	10
Tf2t	3	1	3		1.5	8.5
Adaptive	5	1	3	4		13

The diagonal is empty since the agents never play against themselves; the content of the minor of rank 4 is fixed. It derives from the interaction of the agent with fixed rules among them. For example, the Perpetual defector always gains when meeting a Perpetual cooperator and so on. From the adaptive agents, instead, we expect that they are able to modify their strategies: with the Perpetual cooperator, the best strategy is to defect, with Perpetual defector the best strategy is again to defect. With the TfT the correct strategy is to cooperate always, while with the Tf2t , as already explained, the best course of action is to alternate a cooperation with a defection.

Measuring the difference with the theoretical maximum result and the actual one gives us a quantitative index of their effectiveness. In the evaluation of the adaptive behaviour the achievement of the maximum payoff is a sign of successful adaptation. When this is not reached the table still allows to trace the mistakes in the process: if an agent with fixed rules earns more than what was expected it is because its opponent has not been able to adapt its behaviour.

The values in table 2 can be applied to diverse criteria of effectiveness as follow:

1. the stability of rules will be revealed by the a small variation in the average results once the optimal performance has been reached.
2. The skill to remember a high number of strategy will be demonstrated by the limitation of results reached by agents with fixed rules to those set by the table.

3. The possibility to segregate strategies will be witnessed by the ability to maintain results once achieved even in the presence of more than one agent for each kind of player.
4. The ability to re-elaborate strategies which have become sub-optimal can be measured through time (read number of steps necessary to return to optimality after an identity disguise).
5. Ability to recall forgotten strategies is measured as the ability to return to the optimum after a complete turn of identity disguise, that is to say when the agent will start to show their type correctly.
6. Graceful degradation will be defined by the oscillation of the performance of the adaptive agents around the optimal value. This value can be computed by adding to the values in table 2, three units. The best strategy towards a Noise player is to always defect. While if is defect one gets 1 (instead of 0) while if it cooperates one gets 5 (instead of 3). On average (Noise player picks a strategy from a uniform distribution) the payoff is 3.

Experiments

In order to let the GA reach the optimal result we have experimented different codifications of the structures. In practice, the positions of a binary string are grouped in pairs. Each pair is to be applied to a given type of opponent. The first position says what to do if in the previous run, the GA has defected with that kind of agent, while the second says what to do in the opposite case. At the beginning of each tournament, the agent's rule master ask to the rule maker of the agent to compute and then to give the two natural numbers that identify respectively, the type of opponent and the behavior to act.

Golem, the program used for the computation of the GA includes a system of manipulation of the values of fitness that aims at improving the precision of the selection process: prior to each evolution, the fitness of the various agents are linearly transformed as to assign a minimum value to the worst individual and to scale the others. The process of selection takes place on the differences of individuals' fitness rather than on absolute values. When the population starts to converge, the fitness values are very close and so are the probabilities of being selected for extinction or reproduction. It follows that the algorithm loses its precision and can converge on local optima. Operating on differences in fitness, convergence is delayed to the advantage of the efficiency of computation [11].

We have run simulations on the basic scenario with the addition of the Noise player. For each step of the model two round robin tournaments have been played. In all the simulations the GA has obtained a higher payoff than the TfT. This confirms our idea that an adaptive agent can behave in a fitter way than an agent with a fixed behaviour does.

Codification of the GA's individuals

Three different codification of knowledge have been tested together with strings of different lengths. Invariant elements of the tests have been:

- Each string's value represented an action: zero for defection and one for cooperation, respectively.
- The rule's fitness was represented by the sum of the payoffs accrued to the agent after the double round robin tournament.
- Agents were identified by a number that characterises their type: 0 for Perpetual cooperator, 1 for Perpetual defector, 2 for tit for tat, 3 for tit for two tat, 4 for Noise player and 5 for GA.

The first codification implied individuals represented by strings of 16 positions. The position to be used was determined on the ground of the outcome of the last four interactions with a given type. In practice, a sequence of four 0 or 1 can be read as binary number. For example, to a sequence of four cooperation ('1111'=15) the GA answered by using the value container in the 16th position. The algorithm has not given satisfactorily results since the histories of a Perpetual cooperator and of a TfT with which the GA had cooperated looked equal therefore getting the GA confused.

In order to differentiate the histories their composition has been varied. They have been expressed as the combination of the action of the GA and the action of the opponent in the last two turns. This codification too was not leading to good results.

The best formalisation has been that in which two positions were used for each kind of opponents. The choice of the right pair was determined by the kind of agents, while the choice concerning the element of the pair depended on the last action performed against the opponent. Defection (zero) pointed to the first value, while cooperation (one) pointed to the second.

Simulation 1: a GA against a Perpetual cooperator, a Perpetual defector, a TfT and a Tf2t

The following table reports the values referred to the performance of the agents. The column shows the average performance of the agents in 100 steps (that is to say in 20000 round robin tournaments). The column "GA_steps" shows the number of tournaments played before reaching the optimal result. The average performance of GA is computer after that step, that is to say after that the adaptation has taken place. In the data we have not considered simulation in which the GA had not reached the optimal payoff which amount to about the 8% of the total. Since for each step two tournaments were played the optimal result in table 2 must be multiplied by two, it therefore amounts to 26 for each step of the simulation.

Table 3: summary of results (100 simulations GA, Perpetual cooperator, Perpetual defector, TfT, Tf2t)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*GA_steps*
Average	12,24281123	16,83322732	19,51304236	16,88418239	25,24145993	4135
Variance	0,191150471	1,096997034	0,015980348	0,001643844	0,905137819	1419811
Max	15,618062	22,070208	19,740974	17,011702	25,640125	9502

Table 3 provides us with interesting insights, even if, the preliminary character of the present work suggest to be wary. In particular:

1. The general effectiveness of the GA is confirmed by its ability to reach – in about 90% of the simulations – the optimal payoff and by its ability in over performing the TfT in the entire set of simulations. This result is confirmed by the analysis of the maximum values.
2. The variance exhibited by the distribution of the number of steps to the optimum

seems to suggest a link between efficiency in computation and luckier distributions of pseudo random numbers

3. The small differences between average result after the optimum and optimal results seems to signal satisfactorily stability of the produced rules (effectiveness criterion sub 1). Such results are obtained in spite of the fact that the process of adaptation remains active, as the small differences (due the action of the mutation genetic operation) show.
4. It also emerged that the GA has been able to face four different types of agent and to recall a high number of strategies (criteria sub 2).

If the case in which the optimum is not reached the results are:

Table 4: (100 simulations GA, Perpetual cooperator, Perpetual defector, TfT, Tf2t)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*
Average	12,26600366	17,06405035	19,41568762	16,88464849	24,53270529
Variance	0,217491293	2,099318732	0,077683818	0,001809061	0,347025004
Max	15,80348	22,268627	19,689569	17,058306	25,242723

Differences in the maximum values do not coincide since random seed have changed and non-optimal results are included. However, even under this setting the primacy over the TfT is confirmed. The average value of performance, in addition moreover, results almost insensitive to the learning phase (between the two performances there is a difference of 0.7).

Simulation 2: a GA against a Perpetual cooperator, a Perpetual defector, a TfT, a Tf2t and a Noise player

As explained above, in the computation of the optimal payoff we must include the defection strategy against the Noise player. In this case the maximum extractable payoff amounts to 32 (26+6). The following table is analogous to the previous one:

Table 5: (100 simulations GA, Perpetual cooperator, Perpetual defector, TfT, Tf2t, and Noise Player)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*Noise Player*	*GA*	*GA_Step*
Average	15,30577858	22,93520161	23,86042712	20,62211818	21,13336238	30,91133023	4882
Variance	0,124678132	0,865648072	0,038801733	0,008649615	0,27003176	0,493050738	2442
Max	18,509151	28,010201	24,140015	21,074808	24,376638	31,583733	9302

From the average values it emerges that the inclusion of the Noise player determines an increase in the performance of the Perpetual cooperator - that benefits from the incidental cooperation of the Noise player, and of Perpetual defector, that systematically defecting plays the best strategy ever. The GA improves its performance of about six units as to witness for its flexibility in managing the situation. Its comparative (with the theoretical maximum) performance remains almost unchanged: 25,24/26 = 0,97 under the first scenario and 30,91/32 = 0,965 under the second. This table does not include the run in which the GA has not reached the optimal performance. The percentage of success (about 86%) has been inferior to the one of the previous scenario. Also the number of tournaments needed to converge has increased from 4135 to 4882. We can conclude that the GA is able to face situations imperfect information pretty well.

Table 6 reports the data with the inclusion of the perfectioning of the strategies:

Table 6: (100 simulations GA, Perpetual cooperator, Perpetual defector, TfT, Tf2t, Noise Player)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*Noise Player*	*GA*
Average	15,27887488	22,95305538	23,72849376	20,63673164	21,31654275	29,89539553

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*Noise Player*	*GA*
Variance	0,087070327	1,086595168	0,116486196	0,00994727	0,811402154	0,429179854
Max	17,822382	27,660967	24,145515	21,002701	25,373337	30,869987

Simulation 3: a GA against 5 Perpetual cooperators, 5 Perpetual defectors, 5 Tfts, 5 Tf2ts

Due to the higher number of players the maximum payoff must be re-calculated by multiplying the case for simulation 1 by 5, the new payoff is 130, (26*5) while for the other agent see the following table:

Table7: optimal results (5 Perpetual cooperators, 5 Perpetual defectors, 5 Tfts, 5 Tf2ts and a GA)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*Total*
Perpetual cooperator	24	0	30	30	0	84
Perpetual defector	50	8	10	10	2	80
TfT	30	10	24	30	6	100
Tf2t	30	10	30	24	3	97
GA	50	10	30	40		130

In this setup the GA has never reached - within the 100000 steps - the maximum payoff for at least 100 consecutive tournaments. We have therefore reported the all the results in a single table. The failure notwithstanding, the GA has performed better than the TfT. In the following table, the agents' performance is computed as the average of the performance of the entire category:

Table 8: (100 simulations a GA, 5 Perpetual cooperators, 5 Perpetual defectors, 5 Tfts and 5 Tf2ts)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*
Average	83,45426636	80,06744659	98,44527959	96,70401344	114,0480098
Variance	0,248170054	10,69763964	4,12596272	1,168798689	5,603900301
Max	83,8754882	83,6775696	98,8884094	97,9229126	117,509148

The impossibility to identify the opponent reduces the information available to the GA and therefore its performance, in terms of effectiveness, decays. This takes place especially in the interaction with the tf2t where the alternation of strategies results greatly undermined by the number of tf2t players. In spite of this difficulty the performance is still better than that of the TfT. The reduced payoff of the TfT (98,8 instead of 100) suggests the presence of phases in which the GA does not cooperate with the TfT. In such a case they would gain only the payoff from defection (2) instead of the payoff from cooperation (6). This effect was not appreciable in the previous simulations that did not include the GA's learning phase

Simulation 4: a GA against a Perpetual cooperator, 3 Perpetual defectors, 5 Tfts, 7 Tf2ts
The following table shows the re-computation of the optimal result:

Table9: optimal results (a Perpetual cooperator, 3 Perpetual defectors, 5 Tfts, 7 Tf2ts and a GA)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*Total*
Perpetual cooperator		0	30	42	0	72
Perpetual defector	10	4	10	14	2	40

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*Total*
TfT	6	6	24	42	6	84
Tf2t	6	6	30	36	3	81
GA	10	6	30	56		102

By means of this simulation, we want to test the ability of the GA in preserving good strategy even under the case of low frequency of application (criterion sub 3). Strategies that are discovered with less numerous instances of agents have a different weight in the population. Results are summarised in the following table in which the performance of each category is obtained as the average performance of its components:

Table 10: (100 simulations a GA against a Perpetual cooperator, 3 Perpetual defectors, 5 Tfts e 7 Tf2ts)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*
Average	72,72511829	41,71784914	82,61127814	80,86680298	82,86439908
Variance	2,687250233	47,174969	9,235931932	0,419699427	16,08235338
Max	77,059402	47,355667	83,1302918	81,44601871	88,682968

In this case the GA has almost lost any advantage on the TfT. We must conclude that the GA encounters some difficulties in preserving non-used strategies. The result is unsurprising since the functioning of the algorithm is based on the convergence of the population on only one type of structure.

Simulation 5: a GA against a Perpetual cooperator, a Perpetual defector, a TfT and a Tf2t disguising their identities

In this simulation we aim at testing the GA's skill in rapidly adapting its old strategies (criterion sub 4) and to recall strategies that regain their optimality due to changes (criterion sub 5). As told above, in order to change the obtained results we swap the identification number among agents. The GA gets ‘confused’ by the fall in the performance of its strategies and therefore the adaptation process starts anew. In order to allow for re-adaptation of strategies the time span has been extended to 30000 steps. Results are shown in the following table:

Table 11: (100 simulations with a GA and ‘mutant’ agents

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*GA_step*
Average	15,68981672	16,55280177	19,79281416	17,96980271	19,58232711	3675
Variance	0,229094825	2,683648783	0,001419646	0,001426937	0,185730196	431771
Max	19,067236	20,062836	19,897797	18,098169	20,609148	4902

As stressed the GA exhibits results which are inferior to those of the TfT. It is possible to infer that the convergence mechanism becomes viscous when it is a matter of rapidly adjusting strategies. The GA has reached the optimal results in about the 70% of the cases (those reported in the table), however, the low average performance reveals a strong difficulty in sticking to it when a crucial information changes.

Simulation 6: A GA with a majority of cooperating individuals against a Perpetual cooperator, a Perpetual defector, a TfT and a Tf2t.

Given that optimal strategy in PD leads to defection we wanted to test whether the GA was able to adapt an extremely unfit population of rules. In order to experiment this attitude the GA has been generated with a population of only 5% of defecting genes. In the previous simulations cooperative and detective genes were allowed with the same probability. Table 12 summarises the results:

Table 12 (100 simulations, cooperative GA)

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*GA_Step*

	Perpetual cooperator	*Perpetual defector*	*TfT*	*Tf2t*	*GA*	*GA_Step*
Average	12,83623466	21,52990094	19,78939303	17,15660957	23,82168744	731
Variance	1,098069134	4,513828297	0,000216368	0,220107177	0,5692942	126059
Max	17,80048	23,884789	19,79398	19,783978	25,253975	2002

The number of steps refers to the achievement of the Tft's maximum payoff (20 for 100 consecutive steps). In spite of the extremely unfavourable initial conditions the GA has reached good results similar to those obtain in normal situations.

Concluding remarks and future works

The run of the different simulations has demonstrated that an adaptive system can easily over perform an agent with fixed rules. The ability to exploit all the available information has allowed the agent to develop optimal strategies with all the different kinds of agent, and to beat or equal the TfT performance by confirming our initial intuition.

On the ground of the simulation output, the GA confirms itself as a good algorithm to generate an effective adaptive behavior. Experimentations have revealed the importance of the codification of knowledge with a special attention to the case in which strings that are lessicographically different can earn the same fitness. When the graphic identity does not correspond to the semantic one the GA would be cheated with a subsequent decay in its performances. Even under uncertainty, with agents that play randomly, the GA has developed good solutions. Difficulties in rapidly adjust its set of rules have emerged under settings that are utterly inadequate, leaving few schemes to elaborate in order to obtain better individuals. It has also expressed a remarkable skill in letting good genes emerges even if they constituted only a small portion of the population.

Future works will extend the setup in these directions:

1. inclusion of further agents with fixed rules.
2. test of games different from the PD.
3. repetition of the listed experiments with a Classifier System and comparison with the GA. Further developments will concern the abandonment of the evolutionary metaphor for the connessionistic paradigm by means of the Artificial Neural Network.

References

[1] Axelrod R. The Evolution of Cooperation 1984

[2] Russel S.J., Norvig P. Artificial Intelligence. A Modern Approach 1995 Prentice-Hall

[3] Tesfatsion L. (1997). Old Web Site for Agent-Based Computational Economics (ACE)
(reperibile su http://www.econ.iastate.edu/tesfatsi/abe.htm)

[4] Langton C., Minar N. e Burkhart R. (1995,) Web Site of Swarm (reperibile su http://wiki.swarm.org).

[5] Terna P. (1998). Creare mondi artificiali: una nota su Sugarscape e due commenti - Sistemi Intelligenti, 3/98, pp.489-496. Bologna: Il Mulino.

[6] Newell A., Simon H.A. (1976), Computer Science as Empirical Inquiry: Symbols and Search - Communications of the ACM, 3, pp. 113-126.

[7] Rich E., Knight K.(1991), Intelligenza Artificiale, Milano, MCGraw-Hill

[8] Goldberg D.E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning , Addison Wesley Longman Inc.

[9] Holland J.H. (1975), Adaptation in Natural and Artificial Systems, Cambridge MA, MIT Press.

[10] Chattoe E. (1998). Just how (Un)realistic are Evolutionary Algorithms as Representation of Social Processes - Journal of Artificial and Social Simulations 1,3 (reperibile su http://www.soc.surrey.ac.uk/JASSS/1/3/2.html)

[11] Ferraris G., Morini M. Simulation in the textile industry: production planning optimization, in Baldini M., De Paoli F, Martelli A., Omicini A. atti del WOA2004 Dagli Oggetti agli Agenti Sistemi Complessi e Agenti Razionali – Cd Pitagora Editrice – ISBN 88-371-1533-4, pp . 143-149

[12] Ferraris G., CW0.1 – Classifier Workbench (contribution on http:\\wiki.swarm.org)