Adaptive Behaviour in the Classical Iterated Prisoner's Dilemma

Bruno BEAUFILS ; Jean-Paul DELAHAYE ; Philippe MATHIEU
SMAC Team
Laboratoire d'Informatique Fondamentale de Lille – UPRESA 8022 CNRS
Université des Sciences et Technologies de Lille – Bât. M3
F-59655 Villeneuve d'Ascq Cédex
beaufils@lifl.fr delahaye@lifl.fr mathieu@lifl.fr

Abstract

In this paper we try to show that adaptation is a necessary behaviour feature to use when cooperation between agents is expected. We work with a widely used formal model coming from Game Theory: The classical iterated prisoner's dilemma. We first describe the model, strategies, which are modelizations of behaviour, evaluation methods and some well known results about it. Then we introduce some new strategies which we have set up. Those new strategies have been built with adaptation in mind. We try to show through some experiments results that they seem to be more efficient as well as more robust than established strategies. Analysing those results leads us to state that adaptation is a complex task to achieve, but that it seems to be a key feature in agent behaviour, cooperation in multi-agent systems and furthermore in intelligence.

1 Introduction

The study of cooperation and evolution of cooperation between agents is very often done through a mathematical model coming from Game Theory. This game introduced by M. FLOOD and M. DRESHER in Flood (1952), has been wide spread after R. AXELROD book, see Axelrod (1984). The Classical Iterated Prisoner's Dilemma (CIPD) is a simultaneous two-person non-zero-sum non-cooperative game where each player has to choose between two moves: C (for cooperation) or D (for defection). Payoff for each possible situation is defined by table 1. To have a dilemma, which stands on the fact that individual interests differ from collective ones, the following inequation has to be respected: $S < P < R < T$. This model was widely used and studied since it is one of the most interesting model of all such games, as stated for instance in Rapoport and Guyer (1966).

Table 1: CIPD payoff matrix. As payoffs are symmetric only row player ones are given. R stands for *Reward for cooperation*, S for *Sucker's payoff*, T for *Temptation of defection*, and P for *Punition of defection*.

	C	D
C	$R = 3$	$S = 0$
D	$T = 5$	$P = 1$

The one shot game is *solved* in Game Theory by the NASH equilibrium which consists of defection from both players[1]. The model is thus extended: In the iterated version players meet each other more than one time, without knowing when the last meeting will occur. The payoff of a player is then simply the sum of each of its meeting's payoff. To favour the cooperation, and also to keep this difference between individual and collective interests the following inequation has to be respected: $S + T < 2R$. Payoff values used in the CIPD are given by table 1.

With such an iterated game, what the opponent did on past moves may influence the way a player will choose its next one. In other word it is then easy to study player's behaviour (called strategies in Game Theoretic words). There are a lot of different ways to setup strategies which may be separated in two main families:

- pure strategies are completely deterministic. What move an agent will choose at a specific iteration is completely *pre*defined according to game history. The behaviour is thus completely computable. Use of random is strictly forbidden.
- mixed strategies may use random distribution of pure strategies. Facing two exactly same situations, *i.e.* two identical game histories, an agent will be able to play different moves. Use of random is allowed.

In all experiments described here we define and use only pure strategies, in order to be able to do exact simulations.

[1]The *solution* is however not PARETO optimal : It is clear that mutual cooperation is more efficient

Here are a short list of some behaviours in terms of CIPD pure strategy:

`all_c` always plays `C`

`all_d` always plays `D`

`tit_for_tat` cooperates on the first move then plays its opponent's previous move

`mistrust` defects on the first move and then plays its opponent's previous move

`per_cd` plays periodically `C` then `D`, let us note (`CD`)$*$

`per_ccd` plays (`CCD`)$*$

`soft_majo` cooperates and then plays opponent's most used move, if equal then cooperates

`prober` plays (`DCC`), then it defects in all other moves if opponent has cooperated in move 2 and 3, and plays as `tit_for_tat` in other cases

`spiteful` cooperates until first opponent's defection, then always defects

`easy_go` cooperates until first opponent's defection then always cooperates

`pavlov` cooperates on first move, then cooperates only if both players played same move on previous iteration.

AXELROD states that in order to be *good* a strategy must:

- be nice, which means do not be the first to defect
- be reactive, which means do not allow an opponent to defect without being punished
- forgive, which means do not punish an opponent forever
- not be too clever, which means to be simple in order to be understood by its opponent

The strategy which is used by AXELROD to illustrate those points is `tit_for_tat`. Since AXELROD's book publication it has become the most used and studied strategy. Its fame has pushed a lot of scientists, from biologists to psychologists, to consider results exhibited by AXELROD as definitive. A kind of theory of cooperation based upon reciprocity has thus been set up in a wide litterature as well as in the mind of a lot of people.

As others, for instance Nowak and Sigmund (1993), Boyd and Loberbaum (1987), we have called into question some of those results, and more precisely the last feature: Simplicity. We do believe that it is possible that *complex* strategies may be better than simple ones, and that there may exist an infinite gradient of complexity in the definition of strategy, each level defining a new criterium of quality.

We do all our work in a discrete case whereas lot of works done on the subject, essentially by biologists, has been done in the continuous case without simulations, since the interest is often focused on population dynamics, see for instance Hofbauer and Sigmund (1998).

2 Evaluation methods

In this kind of study one of the main problem is to evaluate strategies, in order to compare them.

Two kinds of experimentation have been classically used for this purpose. We extended and adapted them in order to do big computed simulations, *i.e.* implying a lot of different strategies in a discrete almost deterministic world.

2.1 Tournament and evolution

The first one is to make a pairwise round-robin tournament between some different strategies. Payoff to each one is the total sum of each iterated game. A ranking is then computed according to the score of each strategy. The higher a strategy is ranked, the better it is. Good strategies in round-robin tournament are well adapted to their environments, but are often not very robust to modifications in the environment.

In order to take into account possible modifications of the environment the second kind of experimentation is a kind of imitation of the natural selection process, and is closely related to population dynamics. Consider a population of N players, each one adopting a particular strategy. At the beginning we consider that each strategy is equally represented in the population. Then a tournament is made, and good strategies are favoured, whereas bad ones are disadvantaged, by a proportional population redistribution. This redistribution process, also called a generation, is repeated until an eventual population stabilisation, *i.e.* no changes between two generations. A good strategy is then a strategy which stays alive in the population for the longest possible time, and in the biggest possible proportion. This evaluation is called ecological evolution.

2.2 Complete classes

In Beaufils et al. (1998) we extend this last method to a much larger scope. We define a *descriptive* method to define strategies, which is less risky than an exhaustive method, which are never objective, nor complete. Some structures (*genotypes*), which can be decoded in a particular behaviour (a *phenotype*), are used. A way to have a strategy, is simply to fill this structure. One way to get a lot of strategies is to consider all the ways of filling this genotype, *i.e.* to consider all possible individuals based on a given genotype. Let us call the set of all strategies described by a particular genotype, the *complete class* of strategies issued by this genotype. One way to evaluate a

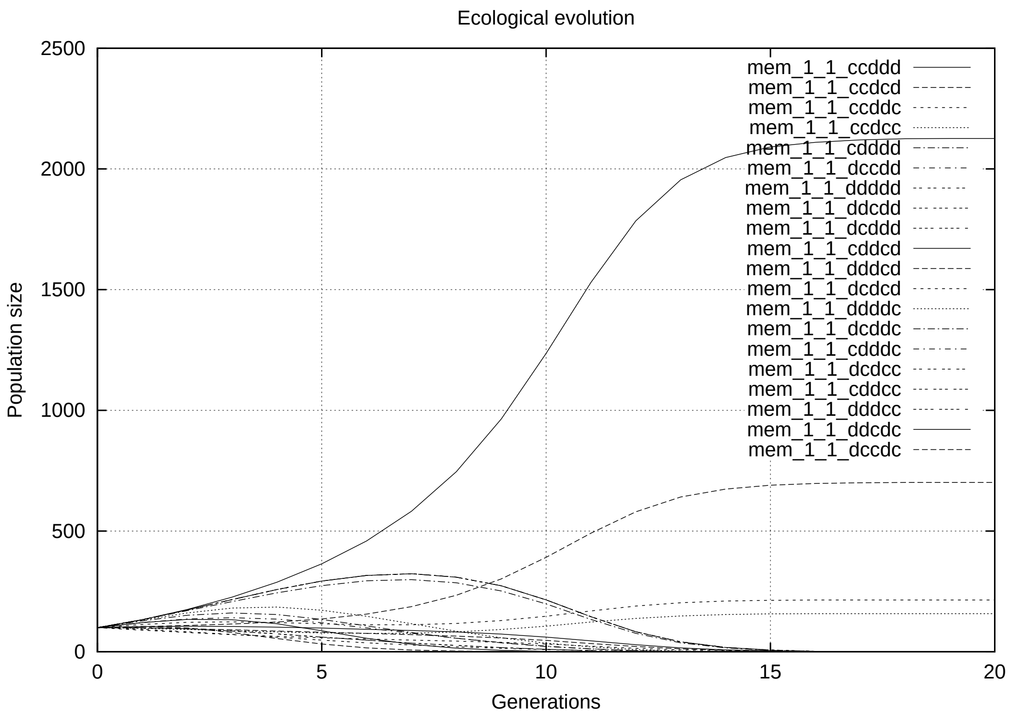

Figure 1: `mem_1_1` complete class evolution. `mem_1_1` contains all strategies using only their last move as well as opponent's last move to play. Only twentieth first strategies are shown.

strategy is simply to make an ecological evolution involving it and all strategies of a complete class. We define some complete classes based on simple ideas like size of strategy memory (how much moves a strategy may remember and use to make its next move). It is then easy to compare two or more strategies using differents complete classes and by comparing their ranking at the end of the evolution ; or to make some complete classes evolution involving differents strategies which could then be directly compared.

Here are a list of some of the families we define and use in our simulations:

- `memory`
  All strategies of this family use only a limited memory of what it played and of what the opponent played. Each strategy of this family is completely defined by the moves to play at the beginning of the meeting and moves to play in each possible faced situations of remembered game history.

  Let us consider, for instance, all strategies which can observe only their last move and opponent's last move. One of those strategies can be defined in this way:
  on the first move I play $\boxed{\texttt{C}}$ then

  $$\begin{cases} \text{if I played } \texttt{C} \text{ and it played } \texttt{C} \text{ then I play } \boxed{\texttt{C}} \\ \text{if I played } \texttt{C} \text{ and it played } \texttt{D} \text{ then I play } \boxed{\texttt{D}} \\ \text{if I played } \texttt{D} \text{ and it played } \texttt{C} \text{ then I play } \boxed{\texttt{D}} \\ \text{if I played } \texttt{D} \text{ and it played } \texttt{D} \text{ then I play } \boxed{\texttt{D}} \end{cases}$$

  The *genotype* of this strategy could be noted as

C	C	D	D	D

  With this genotype $2^5 = 32$ strategies (classical ones included) are defined. The evolution involving all those strategies is shown on figure 1.

  In order to simplify identification, all those informations are embedded in the name of the strategy, which consist of a representation of the strategy genotype. Prefix are used to identify the family and the parameter values. The prefix of the `memory` family is `mem`. The previous strategy is then called `mem_1_1_ccddd` and is the same as `spiteful`.

  Due to early computing limitations with high memory families some variations have been set up. The only modifications are on the way to handle beginning of meeting. Two other prefix are used: `memd` for `memory_with_dynamic_start` family and `memld` for strategies of the family `memory_with_limited_dynamic_start`.

- `binary`
  Strategies of this family use the same concept as those from the `family` but use also a flag which determines if opponents have more cooperated or defected since the beginning of the meeting. The prefix used for `binary` strategies is `bin` and the one used for `binary_with_dynamic_start` is `bind`.

- `moore`
  Strategies of this family are simply MOORE automata. The prefix used for the name of strategies of this family is `moore`.
- `mealy`
  Strategies of this family are simply MEALY finite state automata. The prefix used for the name of strategies of this family is `mealy`.

More details on complete classes may be found in Beaufils et al. (1998) and furthermore in Beaufils (2000).

3 Strategies description

3.1 `gradual`

The first strategy we introduced, in Beaufils et al. (1996), is called `gradual`. It acts as `tit_for_tat`, except when it is time to forgive and remember the past. It uses cooperation on the first move and then continues to do so as long as the other player cooperates. Then after the first defection of the other player, it defects one time and cooperates two times; after the second defection of the opponent, it defects two times and cooperates two times, ... after the n^{th} defection it reacts with n consecutive defections and then calms down its opponent with two cooperations.

As we can see this strategy has the same qualities as those described by AXELROD for `tit_for_tat` and has one more: It is able to adapt its reaction to its opponent ability to understand the sense of the punishment. `gradual` use adaptation to opponent's reaction in time. We have previously shown that this kind of adaptation is really important.

3.2 `bad_bet`

The new strategy we want to introduce with this paper uses another kind of reaction which uses not only time but also space, some kind of generalisation of the idea of adaptation to opponent's behaviour.

The behaviour of this strategy, we called `bad_bet`, is to cooperate until an opponent's defection. As soon as its opponent has betrayed it plays as follow:

1. During 4 moves it plays as `tit_for_tat`
2. then it plays as `all_c` during 4 moves
3. then it plays as `spiteful` during 4 moves
4. finally it plays as `per_ccd` during 4 moves
5. it then compares relative payoff limited to training period for each of those 4 strategies and chooses to play the most interesting one (the one which has given the highest payoff) during next 4 moves.
6. it updates choosen strategy's relative payoff
7. it then goes back to 5

`bad_bet` tries four differents reactions when faced to a non-cooperative opponent, and verify continuously that the reaction it choosed is the best one. Tested reactions correspond to simple punition, complete forgiveness, severe punition and exploitation.

The main idea used here is that `bad_bet` adapts its reaction to opponent's behaviour not only in time (the used behaviour is updated every four moves) but also in space (sequence of behaviour will not be the same against two differents opponent). It is also to be noticed that such behaviour is not as simple as it seems. Whatever approach is taken `bad_bet` is more complex than `tit_for_tat`. This complexity might be illustrated by the time used to compute a move, the memory space used by each strategy algorithm, as well as classical computing complexity. This gives us one more argument to show that simplicity may not be a quality in such a context.

We have found this strategy by conducting a lot of automatic strategy generation tests. The basic idea of our research was to find some combinations of strategy which may be good in a lot of different environments. Based on the framework described above we randomly generate combinaison of strategies and test them, with ecological evolution, in some fixed predefined environments. We then find `bad_bet` and begins to study it. It seems there are no theoretical justifications of the four used strategies. It only seems that among all set of four strategies we have tried this one is a good one. The set can be improved as we will show at the end of the paper. Results presented here are thus purely experimental ones.

4 Some results

We have conducted a lot of differents simulations involving classical strategies as well as ours. Some interesting results of those work tend to show that adaptation is a good quality criterium which may at least be added[2] to those listed by AXELROD.

4.1 Tracking adaptation

First we tried to verify that `bad_bet` really uses adaptation. We would like to be sure that all strategies are sometimes used as reaction. Thus we tracked the behaviour of the strategy during some meetings. Reactions are only used with aggressive strategies.

We conducted a lot of tracking experiments. Table 2 contains examples of some of those tracking results for meeting lasting 100 moves and involving strategies defined in the first section. First column gives the name of the opponent, second gives the date of the first opponent's defection, then comes the used reaction for each four moves period.

[2] and at most can be substituted

Table 2: Tracking bad_bet behaviour. Strategies used by bad_bet are 0=tit_for_tat, 1=all_c, 2=spiteful and 3=per_ccd

Opponent	Date	Played strategies	
all_d	1	0123	000000000000000000000
mistrust	1	0123	111111111111111111111
per_ddc	1	0123	002222222222222222222
per_ccd	3	0123	002222222222222222222
per_cd	2	0123	222222222222222222222
per_cccdcd	4	0123	20003332222222222222
per_ccccd	5	0123	02222222222222222222
prober	1	0123	000000000000000000000
easy_go	1	0123	333333333333333333333

It is easy to see that the best possible reaction is often chosen. For instance against any periodical strategies bad_bet finally chooses to use spiteful, which in this case is equivalent to playing all_d, which is definitively the best behaviour to adopt against such strategies.

Another example is against easy_go. It also chooses the best reaction. Since the opponent can be easily exploitable it uses the most aggressive reaction.

Some other experiments using mixed strategies have also shown that time adaptation is also used. bad_bet is able to change a chosen reaction if the opponent try to change its behaviour during the meeting.

4.2 Adaptation strength

All simulations described here were done using one of our simulator, downloadable on the web. Each meeting last 1000 moves. Each evolution begins with 100 individuals of each used strategies and last until a population stabilisation. Payoff values are those of the CIPD.

Once again we have conducted a lot of experiments but can present only some of them here. We choose the results presented here because they can be considered as representative of all results we have obtained. The main changes between all driven experiments are in the use of differents environments.

Table 3 and figure 2 for instance show the behaviour of bad_bet in a tournament and an ecological evolution involving all strategies described previously.

Figure 3 shows one complete class evolution involving three strategies we have described. Evolution of only the 20 best strategies is represented. It is to be noticed that the complete class called mem_1_2 is the set of all strategies which remember only one move of its past, and two moves of its opponent's past. There are 1024 differents such strategies.

We could notice that in this example the best strategy is the one which has the most adaptive behaviour. Another important point to notice is that order of strategy's strength is exactly order of strategy's adaptive reaction capacity.

Table 3: Tournament involving some strategies and bad_bet. Each meeting last 1000 moves.

```
========== TOURNAMENT

TOURNAMENT RANK
           1 :          bad_bet = 35262
           2 :      tit_for_tat = 32660
           3 :         spiteful = 32660
           4 :           prober = 32148
           5 :        soft_majo = 30998
           6 :           pavlov = 30579
           7 :           per_cd = 28127
           8 :         mistrust = 27176
           9 :            all_d = 26716
          10 :          per_ccd = 26393
          11 :          easy_go = 25517
          12 :            all_c = 24504
Simulations last 0 seconds.
```

Table 4: Results of bad_bet in some complete classes. Gray cells means bad_bet is below the twentieth position.

Class	Size	bad_bet
mem_0_1	8	1
mem_0_2	64	1
mem_1_1	32	1
mem_1_2	1 024	1
mem_2_1	1 024	▨
memd_0_2	128	1
memd_1_2	2 048	1
memd_2_1	2 048	▨
memld_0_3	4 096	1
bin_0_1	32	1
bin_0_2	1 024	1
bin_1_1	512	11
bind_0_2	2 048	1
moore_0_1_2	128	1
moore_1_1_2	2 048	1
mealy_0_1_2	1 024	1

Finally table 4 shows the strength of bad_bet in a lot of different complete classes, since it almost ends at the first position after the evolution is completed. First column gives a description of the complete class used by giving the prefix used in the name of strategies. Second column gives the size of the complete class, *i.e.* the number of defined strategies in the class. Finally third column gives the rank of bad_bet at the end of an evolution involving all strategies from the class and bad_bet. Gray cells means that bad_bet finished below the twentieth position. In each case the *bad* position of the strategy may be explained by the nature of the environment composed by strategies remembering their two last moves and only

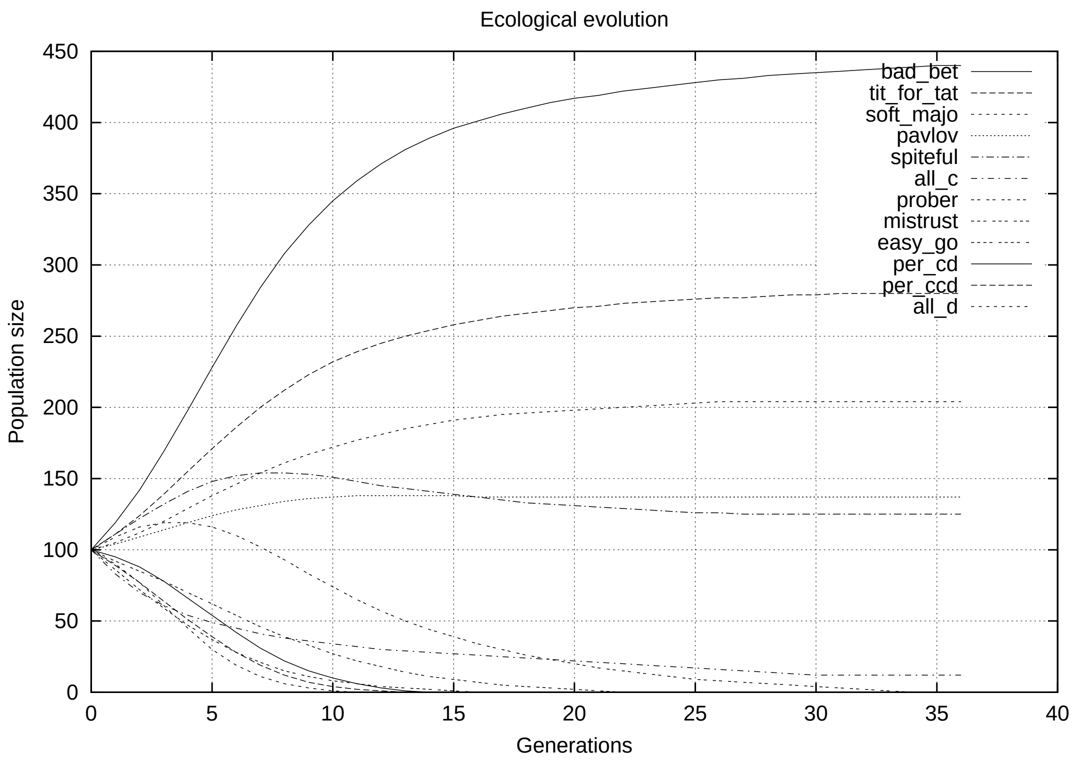

Figure 2: `bad_bet` evolution in a fixed environment. Each meeting last 1000 moves, each initial population sized 100.

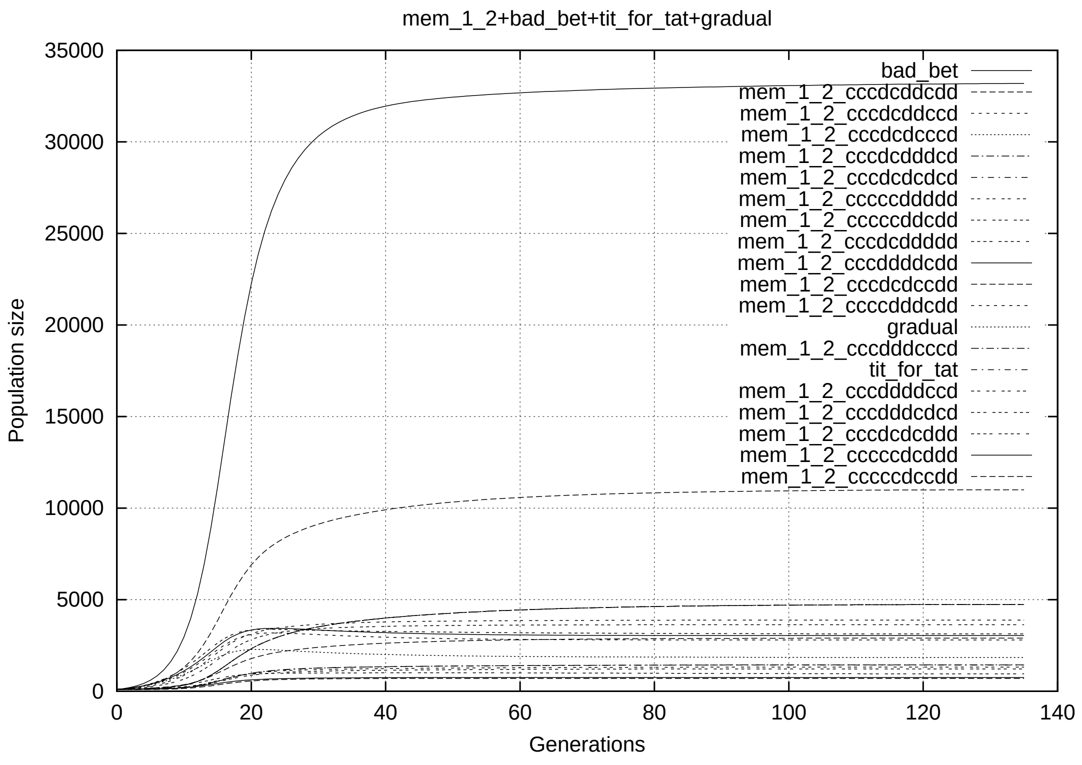

Figure 3: `mem_1_2` complete class evolution involving `tit_for_tat`, `gradual` and `bad_bet`

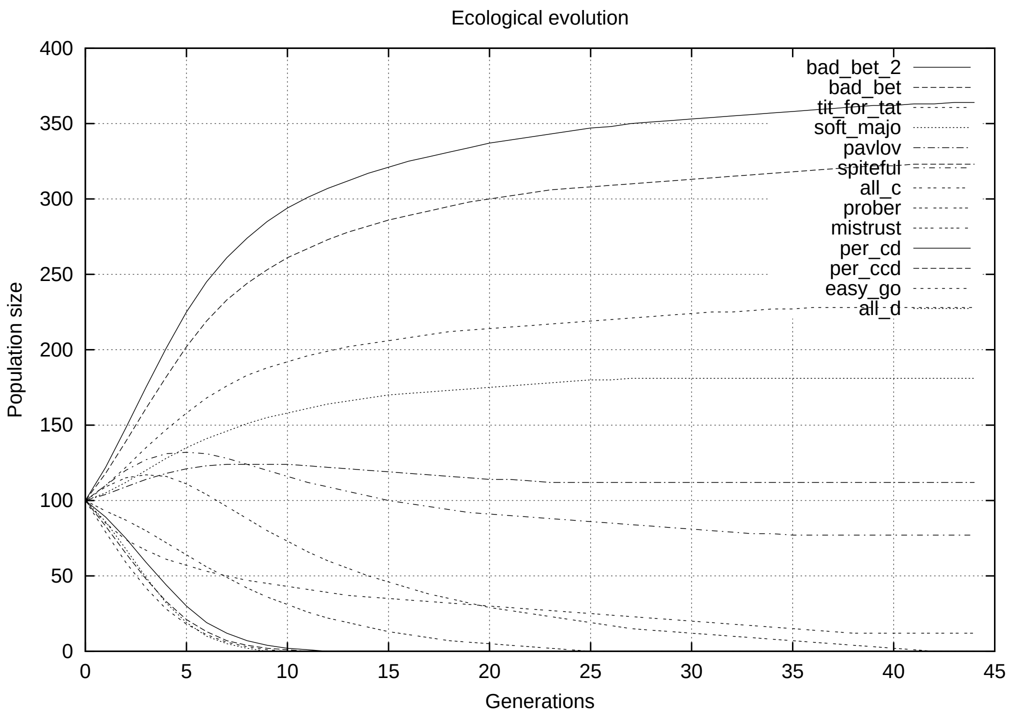

Figure 4: Modified version of `bad_bet` outperfoms all described strategies.

the last move of their opponent. This environment allows some strong combinations between strategies which are aggressive on first moves. Being aggressive at the beginning is not so important for them since they do not take care of it as much as `bad_bet` does. Those two cases apart it is to be noticed that `bad_bet` is a robust strategy which is *good* in a lot of different environments.

4.3 Results analysis

What seems to emerge is that `bad_bet` tries to adapt its behaviour to its opponent's one. Its special kind of adaptation is used by few strategies.

It may be said that, for instance, `prober` adapts its behaviour: If its opponent is weak then it exploits it, else it try to be secure playing as `tit_for_tat`. The main problem is that in order to adapt its behaviour to its opponents it tries to know what behaviour is used by it: `prober` is aggressive.

On the other hand `bad_bet` adapts its behaviour only when it is useful, that is only when its opponent seems to be aggressive. It punishes its opponents trying to determine why it has defected. `bad_bet` does not consider all defection as a tentative of exploitation but consider a defection as a way used by the opponent to try to communicate.

Finally it seems easy to describe each of the four reaction used by `bad_bet`:

- **simple punition**. In this case it considers its opponent understand punition well and that those punitions are interesting enough not to lose two many points.
- **complete forgiveness**. In this case it forgets some opponent's defection because it considers too expensive trying to force opponent to cooperate.
- **strong punition**. In this case it understands the opponent is too aggressive or too clever and would rather stop any kind of cooperation.
- **exploitation**. In this final case it tries to exploit opponent's naive behaviour.

The first used reaction must be significantly reactive or else opponent may have not understood that `bad_bet` is trying to communicate. Then the three others are ordered from the less to the more aggressive one.

The size of the period used to test reactions has to stay short since strategies used by `bad_bet` are simple and may prove their strength on a very short number of iteration. It seems that with larger period speed of reaction is less important than the reaction itself such that reactions are finally not differentiable enough over time.

Having understood that we tried to validate our interpretation by replacing `per_ccd` by `all_d`, which is really a pure exploitation strategy, and have obtained even more better results as shown on figure 4.

It is to be noticed that in order to be good you do not have always to forgive, it depends on your opponent. You must be kind, you must react, and finally you must adapt your reaction to the situation. It may come that the best reaction could be forgiving but that is not a rule.

It seems adaptation means complexity mainly since trying to find how the other behaves, is clearly not a simple task in average.

5 Conclusion

By introducing a new strategy for the CIPD, and some very large computed simulations, we try to show that reaction to opponent's behaviour may be adaptive in order to obtain good results in the CIPD. Adaptation must not be limited to one dimension such as time or space, but has to be as large as possible. Adaptation seems to imply complexity.

A good strategy for the CIPD, which may be a formal representation of an efficient behaviour in an heterogeneous multi-agent systems, may thus have the following features: *Kindness* and *adaptive reaction* (which may include forgiveness). Adaptation may include differents behaviour chosen in a very large scale of reactions. Cooperation may then appear to be one kind of a emergent behaviour in a world where agents are able to adapt to each other.

In a more large sense it will not be completely stupid to say that the faculty of adaptation may be one of the main feature of intelligence in agents world.

Simulation softwares with many strategies as well as informations on the CIPD are available through the web site of the PRISON project located at the following URL:

`http://www.lifl.fr/IPD`

References

Robert Axelrod. *The evolution of cooperation*. Basic Books, New-York, USA, 1984. ISBN 0-465-02121-2.

Bruno Beaufils. *Modèles et simulations informatiques des problèmes de coopération entre agents*. PhD thesis, Université des Sciences et Technologies de Lille, 2000.

Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual, a good strategy for the iterated prisoner's dilemma. In Christopher G. Langton and Katsunori Shimohara, editors, *Proceedings of Artificial Life V*, pages 202–209, Cambridge, MA, USA, 1996. The MIT Press/Bradford Books. ISBN 0-262-62111-8. Artificial Life V, Nara, Japan, May 16-18 199.

Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Complete classes of strategies for the classical iterated prisoner's dilemma. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionnary Programming VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 33–41. Springer-Verlag, 1998. ISBN 3-540-64891-7. Evolutionnary Programing VII, San Diego, CA, USA, March 25-27, 1998.

Robert Boyd and Jeffrey P. Loberbaum. No pure strategy is evolutionarily stable in the repeated prisoner's dilemma game. *Nature*, 327:58–59, 1987.

Merrill M. Flood. Some experimental games. Research Memorandum RM-789-1, The RAND Corporation, Santa-Monica, CA, USA, 1952.

Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998. ISBN 0-51-62570-X.

Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364:56–58, july 1993.

Anatol Rapoport and Melvin Guyer. A taxonomy of 2x2 games. *General Systems*, 11:203–214, 1966.