Experimental criteria to identify efficient probabilistic memory-one strategies for the iterated prisoner's dilemma

Philippe Mathieu, Jean-Paul Delahaye

Experimental criteria to identify efficient probabilistic memory-one strategies for the iterated prisoner's dilemma

Philippe Mathieu[a], Jean-Paul Delahaye[a]

[a] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL (équipe SMAC)*
Centre de Recherche en Informatique Signal et Automatique de Lille,
F-59000 Lille, France

Abstract

Many works and articles about probabilistic strategies for the prisoner's dilemma have already been realised. Notably Press & Dyson 2012 article has lead to renewed interest in the subject. In this article, with the help of a systematic study of probabilistic memory-one strategies, we show that there is a basic criterion to configure and anticipate their success. This criterion, identified through the study of large homogeneous sets of strategies, is then compared to other similar criteria. Our experimental method has allowed us to discover new strategies that are efficient not only in probabilistic environments, but also in more general, probabilistic or non-probabilistic environments. We test the robustness of our results by various methods and compare the new strategies obtained with the best strategies currently known.

Keywords: Iterated prisoner's dilemma, probabilistic strategies, Simulation, Agents' strategies, Evolutionary game theory, Behaviour

*Corresponding author
Email addresses: philippe.mathieu@univ-lille.fr (Philippe Mathieu), jean-paul.delahaye@univ-lille.fr (Jean-Paul Delahaye)

preprint Simulation Modelling Practice and Theory Volume 97, December 2019, pp 101946
Click here for ScienceDirect

1. Introduction

Following the publication of Press and Dyson's article [1] on what they called ZD strategies and the extortion principle, and following the reactions sometimes critical [2, 3, 4, 5, 6, 7, 8, 9, 10] to their conclusions, a great deal of interest has been focused on probabilistic strategies playing the iterated prisoner's dilemma [6, 11, 12]. Yet, no systematic and exhaustive sorting methodology was used to determine whether basic strategies could match or outperform the best known strategies for this problem [13, 14, 15]. We are conducting here this study by combining two of the methods that we consider most likely to produce robust and non-subjective results: the evolutionary (ecological) competition method [16, 17, 13, 18], and the complete classes method [19, 20]. It should be noted that the competitions carried out in this article are synchronous like most of the work in the field and the seminal work of [16] and unlike other recent works such as [21]. We use complete classes built with probabilistic strategies selected with the most comprehensive homogeneous possible mechanism in the infinite set of probabilistic strategies. Our experiments involve up to 5,000 strategies simultaneously, which is currently a record.

The results we obtain are unexpected: some basic strategies yet unknown emerge among the thousands of strategies put in competition. A fairly large category of strategies is identified as robust and efficient for evolutionary competitions. A parameter denoted `p'` is identified and interpreted; It is correlated with the success of the strategies and seems therefore to provide an efficient criterion for predicting the behaviour of a probabilistic strategy in an evolutionary competition. More complex, but precise variants of this parameter, are sought by a comprehensive statistical exploration method.

Systematic series of tests are carried out to ensure robustness of the obtained results. In particular, we confront new probabilistic strategies identified in some deterministic strategies environments to ensure that they remain efficient outside the context that allowed them to be discovered. Newcomers are also confronted with the strategies identified by Press and Dyson and also with

the best known strategies in the iterated prisoner's dilemma. This leads us to a new formulation of general experimental conclusions about the optimal strategies known in the prisoner's dilemma.

Material considerations regarding our calculations. At the moment we are able to take into account a maximum of 5,000 families (which implies matrices of 5,000*5,000). The number of individuals in each family does not matter since it results from the calculation formulas. Beyond this size, computing and memory capacities become too large for a standard desktop computer (Macbook Pro , Intel Core i9, 2,9Ghz, 16Go Ram. Programming language Java). 5,000 families takes us several hours of computation, which is the limit of what we have set for our experiments.

2. Definitions, rules

The prisoner's dilemma [16, 22, 18, 23] is when two entities have the choice between cooperating (`c`) or defecting (`d`) and which are remunerated by `R` points if they both play `c`, by `P` points if they both play `d`, and receive respectively `T` and `S` points if one plays `d` and its opponent plays `c`. We usually describe the rules using the following notations: `[c c]`→`R+R`, `[d d]`→`P+P`, `[d c]`→`T+S`.

For the situation to be that of a dilemma, we impose that [16]: `T>R>P>S` and `T+S<2R`. Usually, the following values are used in simulation or experimental works `T=5`, `R=3`, `P=1`, `S=0`.

In such a situation, defect is a logical behaviour. It always leads to a better result than cooperate. Indeed: (a) if the other entity cooperates, I get 5 points by playing `d` but only 3 points by playing `c`; (b) if the opponent defects, I get 1 point by playing `d`, but 0 by playing `c`. It is a situation of dilemma because collectively the two entities win 6 points by playing `[c c]` while they win less by playing `[c d]` and even less by playing `[d d]`. The collective interest is that everyone plays `c`, but an individual logical analysis inevitably leads to `[d d]` which is collectively the worst case.

The dilemma is iterated when one imagines that the same two entities reg-

ularly have to choose between `c` and `d`. Indeed, to play consists in choosing a strategy that, informed of the previous behaviour of its opponent and of its own behaviour, indicates how to play the next round.

Several studies [19, 13, 23, 24, 25, 26], lead to the following conclusions on which a general agreement seems established.

- (a) There is no one strategy better than all the others. Some are bad in nearly all possible environments, while others are efficient and have success (winning many points) in various competitions.
- (b) Efficient strategies are reactive (they react when they are betrayed), take the risk of cooperating (they begin by cooperating and facing an adversary who cooperates, they do not attempt to defect), and they know how to be indulgent (after a defection of the opponent they forgive in order to renew a cooperation phase. This is for example the case of the `gradual` strategy (see its definition in Appendix).

2.1. Simulation of evolution

In addition to the evaluation of the strategies obtained by organizing various competitions (for example round-robin competitions), test methods exist which simulate an evolutionary process for which only robust strategies succeed [16].

The evolutionary competition method we use, commonly used in literature of the field [16, 23, 27], sometimes called *ecological competition*, is as follows. Several copies of each strategy (e.g. 100) are put into a virtual arena. A round-robin tournament (each strategy meets all strategies) is then organised. Depending on the number of points won during this tournament, the size of each strategy family is adjusted proportionally to the total number of points, which defines a second generation. This second generation produces another generation using the same method, etc. The winning strategies are those with the highest family size. They are usually efficient in various arenas, so their good rankings have a deeper meaning than that given by a simple round-robin tournament. This process is fundamentally different from a genetic algorithm. A

genetic algorithm aims to create new individuals, while a ecological competition only changes the size of populations to identify the most robust.

Here is a formal description of the involved process. Let $S = \{s_1, s_2, \ldots, s_n\}$ a set of n strategies.

Definition 2.1. *Meeting between two strategies. We note* $score_t(s_i, s_j)$ *the points won by* s_i *when it meets* s_j *during a meeting of* t *rounds.*

Definition 2.2. *Round-Robin of a set* S *of* n *strategies. The score of a strategy* $s_i \in S$ *in the round-robin is*

$$g_{t,S}(s_i) = \Sigma_{k=1}^{n} score_t(s_i, s_k)$$

Note that each strategy meets itself. The winner s_{i_0} *is such that* $g_{t,S}(s_{i_0}) = max_{j=1,\ldots,n}(g_{t,S}(s_j))$

For the sake of simplicity we omit for now indices t and S.

Definition 2.3. *Evolutionary process for a set* S *of* n *strategies. For each* $i = 1, \ldots, n$ *, we consider* $pop_i(0)$ *individuals of the strategy* s_i*. It constitutes the generation* 0*. In our experiments we consider that the total number of strategies* $E = \Sigma_{i=1,\ldots,n} pop_i(0)$ *is constant.*
We note $pop_i(g)$ *the number of individuals of the strategy* s_i *at generation* g*. The points won by one individual of the* i *family is then*

$$f_i(g) = (pop_i(g) - 1) * score(s_i, s_i) + \Sigma_{k=1,\ldots,n,k\neq i} pop_k(g) * score(s_i, s_k)$$

Each individual meets then each other individual including those of its own family (but not itself). The total number of points distributed at the g *generation is*

$$total(g) = \Sigma_{i=1,\ldots,n} pop_i(g) * f_i(g)$$

We can now compute the generation $g + 1$ *:*

$$pop_i(g + 1) = E * pop_i(g) * f_i(g)/total(g)$$

In our experiments we consider $t = 1,000$ and for each $i \in \{1, \ldots, n\}$: $pop_i(0) = 100$. The choice to have a stable total number of individuals has no influence on the rankings. It is just a sake of normalization. From the formulas, it can be seen that the bottleneck for conducting such experiments is the computation time of the $score_t(s_i, s_k)$ matrix. Indeed, the evolution of populations no longer requires recalculating this matrix but simply reusing its content with multiplicative factors.

This computation models a natural selection process. The results obtained often confirm (but not always, as we will see) those of round-robin tournaments and increase their contrasts. They lead to a surprising conclusion: except in exceptional cases, the arena ends up being occupied only by strategies that never take the initiative to defect (this is the case of `tit_for_tat` of `gradual` or `pavlov`). After a few generations, the arena is occupied by strategies that only play between themselves `[c c]`. The arena is thus in a state of widespread cooperation.

2.2. Complete classes

To conduct objective and unbiased tests that do not depend on strategies identified as efficient or robust, and to give ourselves chances to discover new and efficient strategies, we use the *complete classes method* [17, 14] which consists in systematically include all strategies with equivalent capacities or functioning on a common abstract principle. We consider in particular the `Mem(X,Y)` classes which group all strategies whose round n depends deterministically on the `X` previous moves that the strategy has played and on the `Y` previous moves the opponent has played. In this way, the list of created strategies is unbiased in the sense that none of them has been chosen or eliminated by the experimenter. A `Mem(1,2)` strategy is therefore defined by the first two moves it plays, then by what it does when the past is for example `[d dc]` (it played `d` on the round $n-1$, and the opponent played `d` on the $n-2$ round, and `c` on the $n-1$ round; in this case there are exactly 8 possible pasts). We denote such a strategy by a name like `mem12_ccCDCDDCDD` with the convention that the sequence designates the first

2 moves (in lowercase) and then the answer for the 8 possible pasts taken in the lexicographic order `[c cc] [c cd] [c dc] [c dd] [d cc] [d cd] [d dc] [d dd]` (see Fig 1). The number of possible `Mem(1,2)` strategies is $1024 = 2^{10}$. More generally the size of a `Mem(X,Y)` set is $2^{max(X,Y)}.2^{2^{(X+Y)}}$.

I Play First: c, c

Me-1	She-2	She-1	
C	C	C	C
C	C	D	D
C	D	C	C
C	D	D	D
D	C	C	D
D	C	D	C
D	D	C	D
D	D	D	D

Figure 1: Genotype of a Memory(1,2) strategy. Here the mem12_ccCDCDDCDD

2.3. A selection of 21 strategies

In the rest of this paper we will use among others the set `Select` of 21 strategies derived from [14] which can be considered as containing the simplest strategies mixed with the best strategies identified today (see Appendix and Fig.2).

Finding strategies that outperform or just rank well when added to `Select` is a difficult challenge.

3. Press and Dyson results

3.1. A theoretical breakthrough

The well-known paper of William Press and Freeman Dyson [1] has a provocative title: *"Iterated Prisoner's Dilemma contains strategies that dominate any*

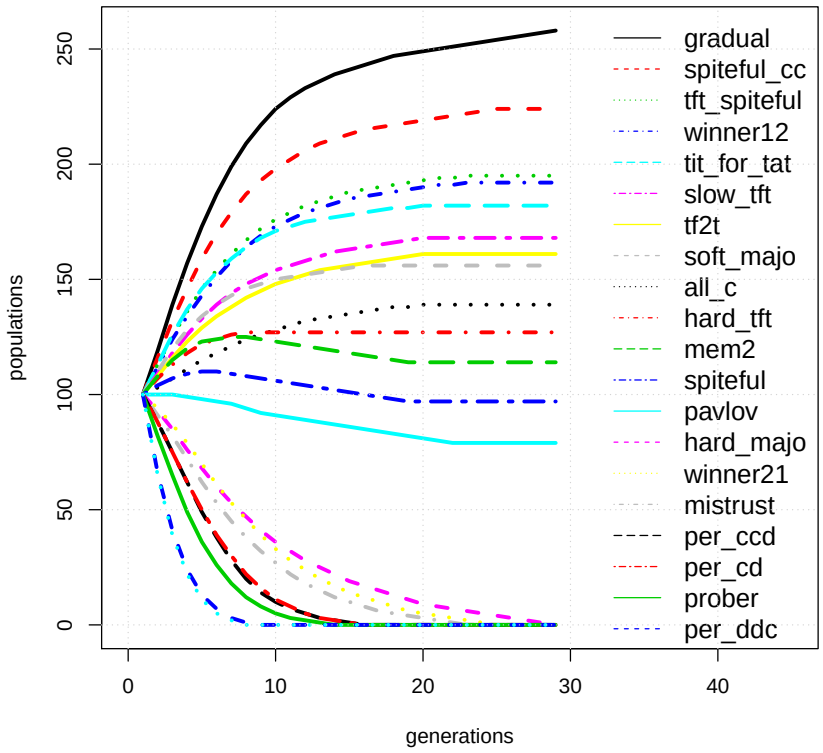

Figure 2: Evolutionary competition of the `Select` set of 21 strategies

evolutionary opponent". It surprised the experts who thought that the agreement on the idea of a convergence towards widespread cooperation prohibited by strategies exploiting the others. Press and Dyson's results thus appeared as a breakthrough in the field. Press and Dyson have discovered that within the 4-dimensional space of memory-1 strategies, there is an interesting 3-dimensional subspace of so-called zero-determinant (ZD) strategies. A player with such a ZD strategy guarantees that his own payoff and the co-player's payoff will satisfy a linear relationship, no matter which strategy the co-player chooses. The space of ZD strategies itself consists of several interesting subclasses of strategies subsequently presented.

Unfortunately, the Press and Dyson reasoning discusses only the average gains of probabilistic strategies when meeting one against one. A simple question always arises and the mathematical arguments do not answer it: among a set as unbiased as possible of probabilistic strategies subjected to an evolutionary process by selection, which probabilistic strategies emerge and win ? That is the issue we deal with in this paper.

Our conclusions reinforce other previous conclusions since the Press and Dyson's paper [2, 3, 4, 5, 6, 7, 8, 9, 10] but, thanks to our systematic method, we succeed here in putting forward a series of robust and efficient strategies that had not been extracted from previous experimental results and, furthermore, we show that a basic criterion exists to identify them quickly.

3.2. Probabilistic memory-one strategies

Press and Dyson's paper proposes two theorems. The first theorem concerns the iterated dilemma in a version limited to probabilistic strategies using a memory-one strategy: the `random`, `tit_for_tat` and `pavlov` strategies belong to this category, but not the `gradual` strategy which, to take its decision, looks at all the rounds already played (full past).

A memory-one strategy is defined by 4 parameters `p1`, `p2`, `p3`, `p4` which indicate the probability of playing `c` when the last round was `[c c]`, `[c d]`, `[d c]` or `[d d]`. Let us note by `proba(p1,p2,p3,p4)` this general strategy. It does not specify how the first round is played, but it does not matter for the mathematical result that does not depend on it. In practice for simulations, we will consider all the strategies whose first play is `c` and all whose first play is `d`.

For example the `tit_for_tat` strategy is coded by `proba(1,0,1,0)`: it cooperates with a 100% probability if the last round was `[c c]` or `[d c]` and cooperates with a 0% probability if not. Similarly, the `random` strategy is coded by `proba(`$\frac{1}{2}$`,`$\frac{1}{2}$`,`$\frac{1}{2}$`,`$\frac{1}{2}$`)` and the `pavlov` strategy is coded by `proba(1,0,0,1)`.

Press and Dyson consider a particular class of `proba(p1,p2,p3,p4)` strategies depending on three parameters `a`, `b` and `c` denote ZD. We will note them `ZD(a,b,c)`.

3.3. The ZD strategies.

The general equations linking the parameters `p1`, `p2`, `p3`, `p4` for the ZD strategies with `R=3`, `S=0`, `T=5`, `P=1` are:

```
p1 = 1+3a+3b+c
p2 = 1+5b+c
p3 = 5a+c
p4 = a+b+c
```

Press and Dyson show that when a strategy `ZD(a,b,c)` is compared to a probabilistic strategy with a memory-one strategy `proba(p1,p2,p3,p4)`, and if we denote `G1` the average gain per round of the first and `G2` the average gain per round of the second, then these average gains satisfy `aG1+bG2+c = 0`. Gains are linearly related. When they meet together, `proba(p1,p2,p3,p4)` is somehow controlled by `ZD(a,b,c)` which means that the ZD will impose the other's gain.

3.4. The equalizer strategies

When `a=0` and `b`$\neq$`0` then `G2=-c/b`. In other words, any memory-one probabilistic strategy has an average gain independent of the probabilities that define it, which depends only on the strategy `ZD(a,b,c)` that faces it. Such a ZD strategy is called an *equalizer*. Relationships become:

```
p1 = 3b+c+1
p2 = 5b+c+1
p3 = c
p4 = b+c
```

and then `G2=-c/b`.

Against such a strategy, all memory-one probabilistic strategies get the same average gain that is known in advance: `-c/b`. There is no need to struggle with an equalizer strategy you will win `-c/b` and no more. The possible values for `-c/b` are all the values between `P` and `R`. Here are the results of the meetings between some known strategies and an `equa` equalizer which is `ZD(0,-1/3,2/3)` which is equivalent to a `proba(2/3,0,2/3,1/3)`, so `G2 = -c/b = 2`. It forces its opponent to obtain and average gain of 2.

```
equa = 2      vs   tit_for_tat = 2
equa = 1      vs   gradual = 2
equa = 11/3   vs   all_c = 2
equa = 3/4    vs   all_d = 2
equa = 2.925  vs   per_ccd = 2
equa = 3/4    vs   spiteful = 2
equa = 2      vs   equa = 2
```

See Annexe 6 for the definition of these strategies. As can be seen, `equa` forces the average gain of the opponent, but this is sometimes done at its own expense, and for example, against `spiteful`, `equa` gets only one point on average per round. Note also that if an equalizer forces the strategies encountered to have a low score it will be its own victim when it plays against itself.

The notion equalizer strategies notion had already been presented in [28] but did not attract attention and moreover are not cited by Press and Dyson.

3.5. The extortioner strategies

Among the ZD strategies discovered by Press and Dyson, some of them operate a kind of extortion. Indeed, if `c=-(a+b)P` (so `a+b+c=0` with `P=1`) one proves that the mean gain `G1` of the ZD strategy against another one (obtaining an average gain of `G2`) satisfies `G1-P=X(G2-P)` with `X=-b/a`.

In short, if the second wants to earn more, and then increase `(G2-P)`, this mechanically implies that the ZD strategy increases its average gain, whose deviation from `P` is always `X` times the deviation to `P` from the average gain of the second one.

The four parameters defining what we will call *extortioner strategies* are given by the equations:

```
p1 = 2a+2b+1
p2 = 4b-a+1
p3 = 4a-b
p4 = 0
```

One of the big flaws of the extortioner strategies is that if `X>1` then they play badly against themselves. If, for example, they want to win twice as much

against their opponent (compared to `P`) it implies that against themselves they will gain only `P`, which is worse than `R`.

If `X>1`, the extortioners will allow you to have a good result only if you give them a proportionally better result. In this case, the extortioners becomes a variant of the `all_d` strategy: no one can beat them, but this means that they also take the risk of a little gain.

3.6. How to recognize ZD, extortioner and egalizer strategies

Practically, if one knows `p1`, `p2` , `p3` , `p4`, in order to know if `proba(p1,p2,p3,p4)` is a ZD strategy, it is necessary to proceed as follows: from `p2 p3 p4` calculate:

```
a =  p2/15 + 4p3/15 -  p4/3 - 1/15
b = 4p2/15 +  p3/15 -  p4/3 - 4/15
c = -p2/3  -  p3/3  + 5p4/3 + 1/3
```

and verify that: `p1=1+3a+3b+c`.

If `a=0` and `b`$\neq$`0` then this ZD strategy is also an equalizer. If `a+b+c=0` then this ZD strategy is an extortioner strategy.

3.7. Utility of a long memory

The second important theorem of Press and Dyson's paper indicates that in a supposedly infinite game, if a strategy A plays against a strategy B having a memory of k rounds, a strategy A' with a memory of k rounds or less exists which obtains the same average score against B. The combination of these two mathematical results of Press and Dyson leads to the assertion that confronted to an equalizer or an extortioner strategy, not only all the strategies with a memory-one strategy are constrained, but all the strategies with finite memory. From this one we are tempted to conclude that: *"(a) Strategies storing more than the last shot are unnecessary. (b) We have, with the ZD strategies, strategies that are dominant for the iterated prisoners' dilemma"*.

Some have surely interpreted the theorems demonstrated in this way, and the title chosen for their paper suggests that this is also the case for Press and Dyson.

Yet the double assertion about the uselessness of extended memory strategies and the superiority of ZD strategies is false. About the usefulness of a long memory see also [29]. Consider first the affirmation of the ZD strategies superiority. It has long been known that in the iterated prisoners' dilemma beating its opponent (getting more points than it) can be done at the expense of the winner and that the latter could have obtained more points on average by agreeing to be beaten.

The `all_d` strategy wins against any other strategy (it is obvious) and for example, in a game of 100 rounds against `tit_for_tat`, `all_d` gets 104 points while `tit_for_tat` wins only 99. The `all_c` strategy does not win against `tit_for_tat` but gets 300 points during the 100 rounds againts `tit_for_tat` who gets also 300 points. When confronted to `tit_for_tat`, `all_d` will win maybe, but it is wrong to win because by doing the same thing as `all_c`, it would have a much better score.

Most of the Extortioners strategies are in the same situation: they force their opponent by renouncing themselves to have good scores. Extortioners win against the strategies they are opposed to, but this is at the cost of the total points earned. Moreover, an extortioner strategy of parameter X with $X > 1$ that plays against itself gets (according to the theory that we verify by simulation) only one point on average per round, which is very weak. It is not true that being a good strategy means beating always your opponent. It is better not to always beat it, get along well with it and get a lot of points.

The case of `tit_for_tat`, which is an extortioner with $X = 1$, is remarkable. We get the impression of a paradox when we state its properties: `tit_for_tat` never beats any strategy individually and is beaten by many strategies, yet it is a good strategy that wins many competitions. It wins not because it forces others to earn less than it does (because $X = 1$), as a real Extortioner strategy (with $X > 1$) does, but because it punishes strategies that do not want to cooperate. It forces cooperation.against it, either you will win few points, or you will cooperate, which will be good for it and for you.

It is therefore a misconception to believe that extortioner strategies are effi-

cient in terms of the number of points earned. In one-to-one meetings, they win against their opponent (like `all_d`), but for that, they hurt themselves, and overall they play rather miserably which is confirmed by [11]. Note that [30] have shown with human volunteers that an additional monetary incentive (bonus) paid to the finally competitively superior player maintains extortion but this is not the case without any bonus.

Besides the error of believing that beating its opponent is gaining points, another oversight leads to the belief that extortioner strategies were superior: to impose oneself one must play correctly against oneself. This is important in round-robin tournaments, but even more in evolutionary competitions. Indeed, if you are the leader during the first generation, the arena will be populated with many strategies identical to you, and you will therefore meet them very frequently. If you play poorly against yourself, it will eventually turn against you. Nothing is false in the mathematical results of Press and Dyson, but by addressing only the problem *"who wins in a one-to-one fight ?"* and forgetting the problem *"how many points are won ?"* and the problem *"Do you play well facing to yourself ?"* the theorems demonstrated do not allow us to conclude that the ZD strategies are efficient strategies. The simulations show without any doubt that ZD strategies are inefficient.

The result of Press and Dyson on the lack of need for a strategy to have long memory is correct : if a strategy A plays against a strategy B which uses a k rounds memory, a strategy A' exists which obtains the same average score against B but which uses only a memory of k rounds. However this does not mean that against two different strategies B and C having a k rounds memory, a strategy A' exists with a k round memory that obtains the same score against B and against C. Indeed, the one A' which can replace A against B, is not necessarily the same as A", which can replace A against C.

To face several opponents, having long memory is useful simply because it makes it possible to distinguish them from others. The result of Press and Dyson on the lack of need of memory is valid only in one-on-one meetings, but is not true as soon as one considers round-robin tournaments or evolutionary

competitions.

Classical simulations confirm that efficient strategies for environments with multiple strategies take advantage of the use of a large memory of the past. On the issues of useful memory or not one can consult [27, 25].

4. Competition among probabilistic memory-one strategies

In a first series of experiments we consider sets as large as possible and homogeneously distributed of probabilistic memory-one strategies, and we confront them using an evolutionary process.

4.1. Massive evolutionary experiments

To obtain sets of probabilistic strategies of the form `proba(p1,p2,p3,p4)`, we set a step of variation of the probabilistic parameters. For `K=5` for example, we make the `pi` coefficients vary in the finite set of values $0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1$ which leads to $2 * 6^4 = 2592$ strategies (the 2 comes from the two possible choices for the initial play). We denote this complete class `ProbaCD_K=5`.

The results for the round-robin tournament are shown below. We used parts of 1,000 rounds and the usual parameters. We calculated the results of each game one by one by making it play 5 times, so as to limit the effects of the probabilistic variations.

rank	strategy identification	cumulated score
1	probaD_0.0_0.0_0.0_0.2	39345809
2	probaC_0.0_0.0_0.0_0.2	39264669
3	probaD_0.2_0.0_0.0_0.2	39244163
4	probaD_0.0_0.2_0.0_0.2	39193409
5	probaC_0.2_0.0_0.0_0.2	39145486
6	probaD_0.4_0.0_0.0_0.2	39136783
7	probaC_0.0_0.2_0.0_0.2	39104825
8	probaD_0.2_0.2_0.0_0.2	39069164
9	probaC_0.4_0.0_0.0_0.2	39043576
10	probaD_0.6_0.0_0.0_0.2	38995942

The first ranked strategy is also denoted `probaD(0,0,0,1/5)` ; The `D` indicates that its first move is `D` (defect); The integer at the end of the line indicates the gain in points in a round-robin tournament repeated 5 times.

In this experiment, the first ZD strategy appears in the 34th rank. This strategy is equivalent to `spiteful`. In the first 100, there are only six ZD strategies. Unsurprisingly, ZDs, extortioners and equalizers are not particularly successful in competitions.

This is not surprising since the criterion that made it possible to identify them only took into account the ability to do better than the opponent in one-on-one meetings, which we know is in no way a guarantee of success in round-robin tournaments (you must win many points) or in evolutionary competitions (you must continue to win many points even when ineffective strategies have disappeared).

For the evolutionary competition, the first ten strategies with their final populations (when stabilisation) are given here:

rank	strategy identification	population
1	probaC_1.0_0.8_0.0_0.0	34262
2	probaC_1.0_0.6_0.0_0.0	31579
3	probaC_1.0_0.4_0.0_0.0	30550
4	probaC_1.0_0.2_0.0_0.0	28640
5	probaC_1.0_0.0_0.0_0.0	27746
6	probaC_1.0_0.0_0.0_0.2	9540
7	probaC_1.0_0.2_0.0_0.2	8893
8	probaC_1.0_0.0_0.2_0.0	8451
9	probaC_1.0_0.2_0.2_0.0	7701
10	probaC_1.0_0.4_0.2_0.0	5984

Note once again that the strategy ranked fifth corresponds to `spiteful`

The evolutionary mechanism used here (Definition 2.3, Fig.3) consists in replacing each generation by a new generation whose populations for a given type of strategy are proportional to the number of points won by those strategies during a general round-robin tournament involving strategies present in

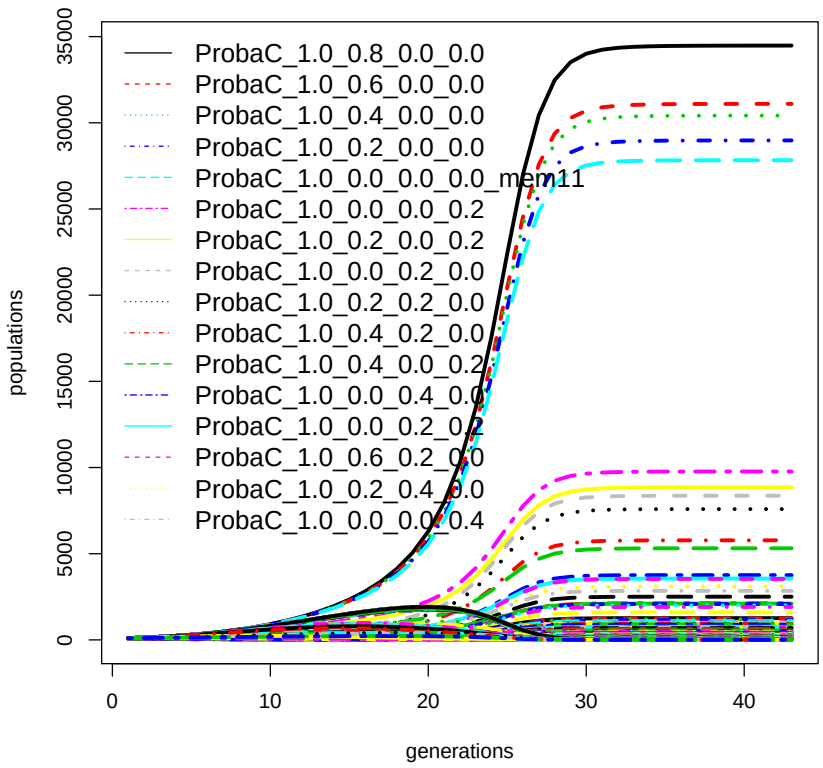

Figure 3: Evolutionary competition of the 2,592 strategies of the `ProbaCD_K=5` family. Strategies that have only 0 and 1 as probabilities are strategies belonging also to `mem(1,1)`. For example here, this is the case of the fifth strategy which is precisely spiteful.

the previous generation: the progeny of a type of strategies at generation n is proportional to the number of points won by the strategies of this type in the round-robin tournament between strategies at the $n-1$ generation. In the beginning, it is assumed that each type has 100 strategies and the total from one generation to the next remains the same (close to rounding problems)

We have also conducted tests when the number of instances of a strategy at the n^{th} generation is obtained by taking a times the number of instances of the strategy at the $n-1$ generation, and $(1-a)$ times the number of instances given

by the previous calculation, with the parameter a between 0 and 1. This models a partial replacement of a generation by the following: the parents do not die right away. The results (ranking and finals) obtained are always very close (if $a < 1$) to those obtained when the replacement of one generation by another is complete. Only the duration of convergence towards the state of widespread cooperation is changed.

As is often the case with this game, the results of a round-robin tournament in a full class that includes many mediocre or bad strategies do not reflect precisely what is found as a result of evolutionary competitions. The reason is simple: strategies that take advantage of the presence of bad strategies in the initial set are quickly downgraded or even eliminated when the bad disappear. To succeed in an evolutionary process, good results must be achieved with those who achieve good results and who are the only ones in the long term to survive. The result of a round-robin tournament in a set with many mediocre strategies has dubious meaning. Only that of evolutionary competitions is relevant.

The winner of the round-robin tournament, with the exception of its last parameter, is the strategy `all_d` ranked 42^{nd}. To be well ranked in the round-robin tournament of the complete class `probaCD_K=5`, it is enough to exploit the mediocre strategies which are very numerous; It is very easy: it is enough to almost never cooperate. That does not teach us anything. Only the result of an evolutionary process that begins with the disappearance of mediocre strategies is of interest.

In the case of evolutionary competition, what is observed seems never to have been noted. The best strategy for this set of more than two thousand strategies is: `probaC_1.0_0.8_0.0_0.0`. The initial size of 100 grew to 34,262 when the state of widespread cooperation was established.

Its behaviour is surprisingly simple: it is a spiteful (since `p3` and `p4` are 0, once it begins to defect, it always defects), but it is a spiteful which reacts without rushing when it is defected: in case of a round `[c d]`, it starts defecting only with a probability of 20%.

In other words, in the initial phase of the game, it cooperates and continues

to do so as long as the other cooperates, and when it is defected in this initial phase, it forgives in 80% of cases. On the other hand, when it decides to defect, there is no longer a possible return to cooperation. We will call these kind of strategies *gentle spiteful*.

The following four strategies are also gentle, but their reactivity in case of defection increases: 40% for the second, then 60% for the third, then 80% for the fourth, then 100% for the fifth, which leads to the usual (`spiteful`).

`probaC_1.0_0.6_0.0_0.0`

`probaC_1.0_0.4_0.0_0.0`

`probaC_1.0_0.2_0.0_0.0`

`probaC_1.0_0.0_0.0_0.0 mem(1,1)`

We note on the plots that these 5 strategies are largely ahead of all the others. The sixth strategy is: `probaC_1.0_0.0_0.0_0.2`. This strategy still has a behaviour that can be interpreted quite easily. It is a spiteful (without any patience since `p2=0`), but, once in its punishment phase, conducts reconciliation efforts: when the round that has just been played is `[d, d]`, it cooperates in 20% of cases, as if it were telling its opponent: *"we have a bad start, I try a first step towards you (in 20 % of cases) to renew a better deal"*.

The following are still susceptible of interpretations to the same type, although more and more complicated. Formally, to be more precise and carry out counts we will call *gentle spiteful* all the `probaC(p1,p2,p3,p4)` strategies with `p1=1` and `p2+p3+p4` $\leq$ `1`. One notes that `tit-for-tat` and `Pavlov` are such strategies. The first 37 strategies in the final composition of the set after stabilization of evolutionary competition belong to this category.

Another remark is that, in evolutionary competition, only 133 strategies keep a non-zero population, and these are all strategies which start by cooperating and which verify `p1=1`. The set has therefore unquestionably converged towards a famous state of widespread cooperation.

4.2. Study of surviving ZDs

The only ZDs that survive are as follows. They are indicated with their rank, their category and their final population (which is never very good):

rank	strategy identification	population
29	probaC_1.0_0.0_1.0_0.0_ZD_Extort	1040
34	probaC_1.0_0.6_0.4_0.0_ZD_Extort	927
35	probaC_1.0_0.4_0.6_0.0_ZD_Extort	910
72	probaC_1.0_0.2_1.0_0.4_ZD	134
125	probaC_1.0_0.6_0.6_0.4_ZD_Equal	5

The first ZD strategy, therefore 29th, is actually `tit_for_tat` which is actually a ZD of coefficient `X=1`. This `X=1` means that in reality it does not extort anything, but forces its opponents to win as much as it does, neither more nor less. This kind of strategy has sometimes been called *generous ZD strategy* [11] and their ability to survive in an evolutionary competition has been identified as much better than those with `X>1`. What we find here confirms that this type of extortioner has some ability to survive, but what we observe also is that they are not the only ones, nor the best.

The second ZD strategy which is 34th, is a ZD with `a=2/25`, `b=-2/25`, `c=0`, `X=1`. It is a ZD of coefficient `X=1` (which like `tit_for_tat` does not extort anyone).

The third ZD strategy which is 35th, is a ZD with `a=3/25`, `b=-3/25`, `c=0`, `X=1`. It is once again a ZD of coefficient `X=1`.

The fourth ZD strategy, 72th of this ranking, is neither an extortioner, nor an equalizer. It is a ZD strategy with `a=2/25`, `b=-7/25`, `c=3/5`. The relationship between the average gain it obtains `G1` and the average gain of its opponent `G2` is `aG1+bG2+c=0`. This leads to : `2G1+15=7G2`. This strategy in the situation of widespread cooperation gets 3 points on average per round, as its opponent.

The fifth ZD strategy, 125rd of this ranking, is a member of the equalizer family with `a=0`, `b=-1/5`, `c=3/5`, `-c/b=3`. This is an equalizer strategy forcing its opponent to win 3 points in average per round, which is also its gain during a widespread cooperation.

One notes that the `Pavlov` strategy (which is not a ZD) survives in position 80 with 113 of population

```
80 probaC_1.0_0.0_0.0_1.0_mem11 113
```

4.3. How to recognize efficient probabilistic strategies ?

We can see that the only extortioners or equalizers that survive are actually strategies that do not extort anything in the strict sense. It is remarkable that they are largely beaten by gentle spiteful strategies which are therefore in this evolutionary context better than the large majority of the strategies proposed by Press and Dyson.

As shown in figure 5, the ranking of the 133 strategies whose final populations do not vanish is directly correlated with the parameter `p'=p2+p3+p4`.

In order to anticipate the success of a probabilistic strategy with a memory-one strategy, the double criterion `p1=1` and `p'=p2+p3+p4` as small as possible is very efficient.

This confirms the remarks made on the mathematical arguments of Press & Dyson. The analysis conducted in [1] by studying the average results of probabilistic memory-one strategies, and by focusing only on forcing and controlling the opponent without worrying about the number of points it costs, does not lead to any criteria for identifying strategies that are truly effective as soon as they are placed in an evolutionary context (or even in the context of classical round-robin tournaments).

4.3.1. Understanding the double criterion

We now propose an interpretation and an explanation of this double criterion. To win an evolutionary competition or only to succeed properly, it is necessary to survive when the widespread cooperation is established, It is then necessary to cooperate with the cooperating strategies, hence the `p1=1`. It is also necessary to be reactive, that is to say not to let oneself be carried out and to adopt behaviour sufficiently severe to encourage the other to cooperate. The

hardest form of reactivity is that of spiteful `p2 = p3 = p4 = 0`. Tempering this hardness is acceptable, if moderate, and can be interpreted as follows:

- (a) Choosing a non-zero value not too large for `p2` means that you do not systematically go into the retaliatory state after a round `[c d]`, but only pass it with a certain probability;
- (b) Choosing a non-zero value not too large for `p3` is to accept with a certain probability after a round `[d c]` to try again to cooperate with an opponent who seems to desire it;
- (c) Choosing a non-zero value not too large for `p4` is to accept with a certain probability after a round `[d d]` to take the first step in order to revive a state of mutual cooperation.

Combining these three forms of temperance in a strategy by adopting small non-zero values of `p2`, `p3` and `p4` is not absurd, provided that the total temperance introduced in its behaviour is not too large, hence the criterion on `p'= p2 + p3 + p4`.

In figure 4 the `ProbaCD_K=5` strategies have been grouped into 7 subsets for which we have computed the average ranking generation by generation. There are the strategies for which `p1`$\neq$`1`, then for those with `p1=1`, those with `p'` in the `[0;1/2]` interval, then `[1/2;1]`, `[1;3/2]`, `[3/2;2]`, `[2;5/2]`, `[5/2;3]`.

We can see that when `p1`$\neq$`1`, beyond the generation 30, the rankings becomes mediocre, then bad. For strategies with `p1=1`, the best values for `p'` are between 1/2 and 1 (slightly better than for `p'` between 0 and 1/2). As soon as `p'>1`, the rankings are much worse than for `p'`$\leq$`1`.

4.3.2. Improvement and variant of `p'`

If we compute the Spearman correlation coefficient between the rank of strategies that end with non-zero populations in the evolutionary competition `ProbaCD_K=5` (there are 133) and the parameter `p'`, we find that:

`Cor(rank,p')=0.8805`

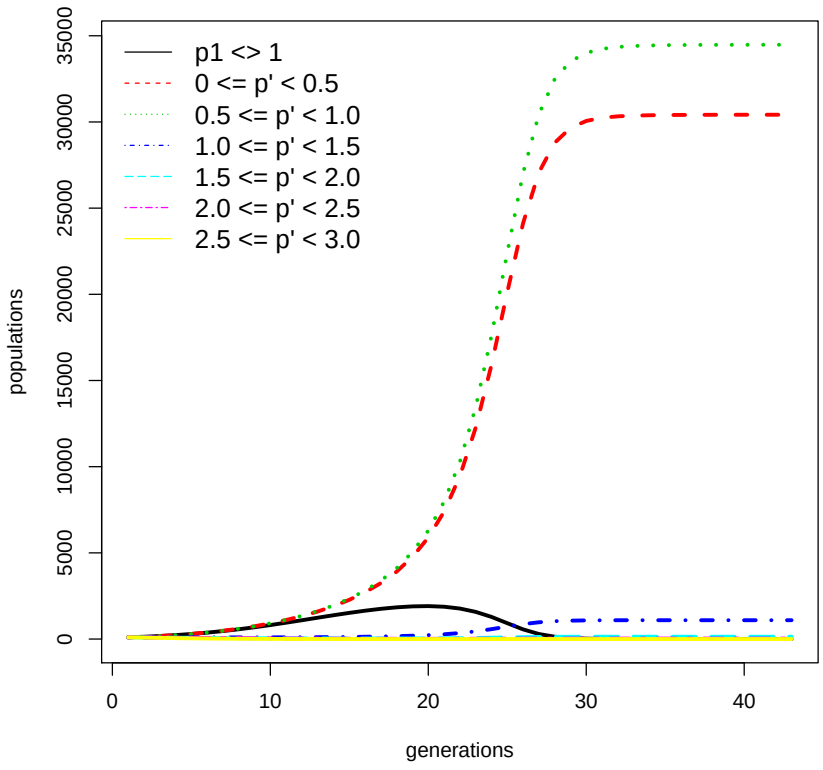

Figure 4: The 2592 ProbaCD_K=5 strategies have been separated into 7 different sets for which we calculated the average ranking generation by generation. It is clear that it is the category corresponding to the interval [0.5 , 1] which is the best. We see again the correlation between p' and the success of a strategy.

which is a very good correlation. We have systematically tried to improve this parameter. The next parameter, which is still very simple, gives a remarkable result.

```
p'' = p2 + 0.5 p3 + p4
Cor(rank,p'')=0.9411231.
```

The optimal still allows a slight improvement:

```
p* = 0.266 p2 + 0.138 p3 + 0.277 p4
Cor(rank,p*)= 0.9413768
```

It should be noted that strategies with `p'` value close to each other, while conceptually very different, succeed in a comparable way. `p'` is therefore a reliable means of anticipating the success of a strategy. The figures 5 and 4 illustrate this correlation.

This result is not unrelated to some theoretical results [31]. This latter proposes as criteria `p1 = 1` and $(T-R)p3 < (R-S)(1-p2)$ and $(T-R)p4 < (R-P)(1-p2)$ but does not correspond precisely to our. Nevertheless, all of the top 10 strategies of 3 satisfy these three conditions.

4.4. Robustness of these experiments

Do the results we have just commented and analysed are robust ? Do they persist when we change the precise parameters of our experience with 2,592 strategies ? That is what we will study now.

In another experiment, we only started with strategies that start with `c` (so there are half exactly), which corresponds to the complete class we call `ProbaC_K=5`) the result is very close: the same 5 first ones are already found with just some permutations of the final ranking.

With `ProbaCD_K=4` we obtain equivalent results: At the beginning of the ranking, the gentle spiteful are in decreasing order of patience (`p2=75%`, `p2=50%`, `p2=25%`, `p2=0`). With `ProbaCD_K=3` it is still the same thing.

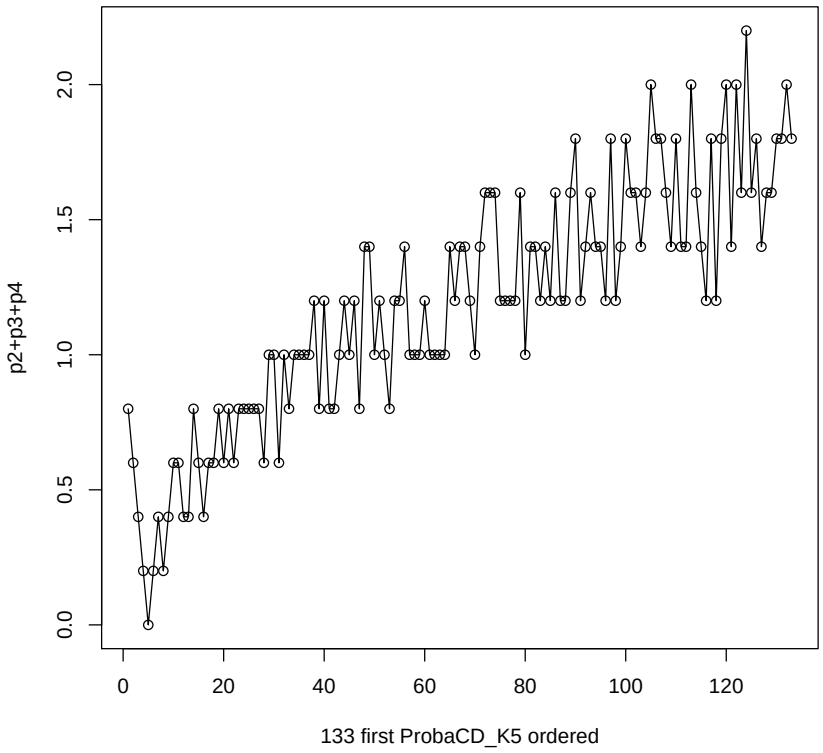

Figure 5: Plot with in x-axis the ranking of the 133 first random strategies and in ordinate the value of p'=p2+p3+p4. There is an obvious correlation between p' and the rank of a strategy.

4.4.1. Objection 1

An objection could be made to our method: the probabilistic strategies composing the initial set are uniformly distributed (making varying the `pi` coefficients in constant steps): this regularity could lead to specific results which would therefore have no general value.

We have then conducted experiments where we chose 2,000 (then 4,000) strategies randomly in the `ProbaCD_K=10` family (which includes $2 * 11^4 = 29.282$ strategies).

The results obtained confirm those of the main experiment with `ProbaCD_K=5`: the winners are in each case strategies of the gentle spiteful family. This is a confirmation of the double criterion.

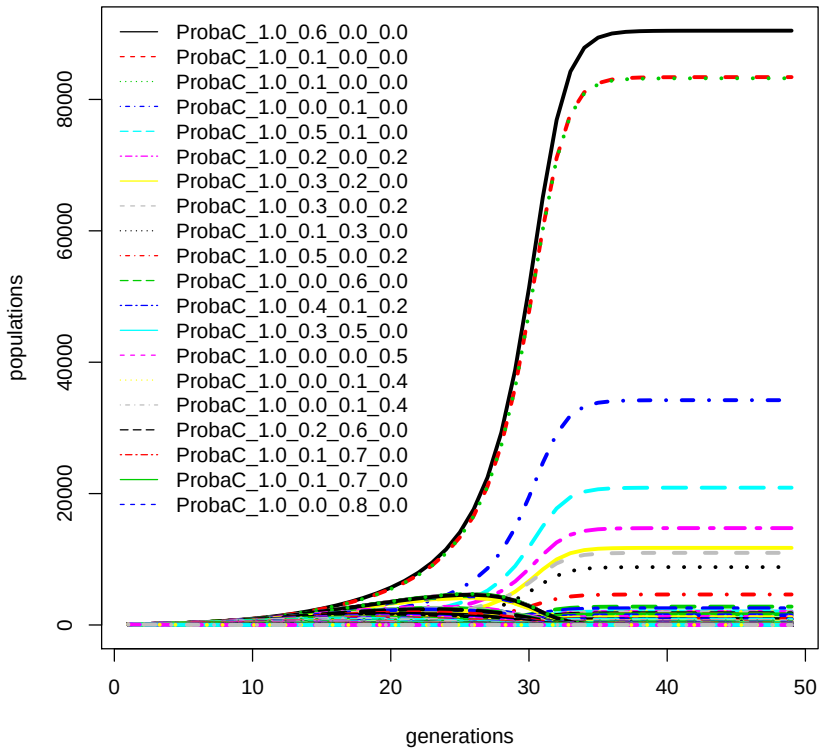

Figure 6: Evolutionary competition of 4000 randomly chosen strategies in `ProbaCD_K=10`. One can note that only subsist strategies beginning with C and with `p1=1` and that the best ones are all gentle spiteful strategies.

Here are some examples. In an experiment with 2,000 strategies we find that only survive 5 strategies:

rank	strategy identification	population
1	probaC_1.0_0.1_0.3_0.0	112317
2	probaC_1.0_0.3_0.4_0.0	42330
3	probaC_1.0_0.3_0.4_0.0	39921
4	probaC_1.0_0.0_0.3_0.1	5077
5	probaC_1.0_0.3_0.3_0.1	354

In another experiment, there are 22 survivors. Here are the first 9:

rank	strategy identification	population
1	ProbaC_1.0_0.0_0.0_0.1	120968
2	ProbaC_1.0_0.1_0.4_0.1	17777
3	ProbaC_1.0_0.5_0.3_0.0	14011
4	ProbaC_1.0_0.1_0.3_0.2	7355
5	ProbaC_1.0_0.2_0.6_0.0	6842
6	ProbaC_1.0_0.4_0.1_0.2	5007
7	ProbaC_1.0_0.1_0.8_0.0	3982
8	ProbaC_1.0_0.2_0.1_0.3	3854
9	ProbaC_1.0_0.3_0.7_0.0_ZD_Extorq	3444
...	...	...

An extortioner appears in position 9 and have a coefficient X which is equal to 1 (it is not, therefore, in the strict sense an extortioner). Note that all the strategies mentioned here are in the gentle spiteful family. The 5 of the first computation, and the 9 of the second computation all have a p'≤ 1. It shows once again that to succeed in this kind of set, what is important above all is not to be ZD, extortioner or equalizer, but to best satisfy the double criterion (or one of its variants with p'' ou p*)

4.4.2. Objection 2

Another objection could be made to our method: we do not consider a sufficient number of ZDs in our initial set. We have taken ProbaCD_K=5 and added a family of 880 ZD (all those whose pi are multiples of $\frac{1}{32}$).

Nothing changes essentially: the first 5 are exactly the same in the same order as for the set ProbaCD_K=5; The first ZD is 25th (with the exception of

spiteful which is 5th) and it is tit_for_tat: probaC_1.0_0.0_1.0_0.0_ZD_Extort.

The next ZD in the ranking is ZD a=0.03125 b=-0.03125 c=0.0_ZD_Extort whose X is 1.

4.4.3. Robustness of new strategies

We wanted to know if the best deterministic strategies identified by [14] obtain also good results in these probabilistic complete classes.

We therefore carried out the computation for ProbaCD_K=5 + Select .

Few things change regarding the relative positions of the probabilistic memory-one strategies but the best of Select are intercalated and take very good ranks.

rank	strategy identification	population
1	spiteful_cc	19286
2	tft_spiteful	19078
3	gradual	18235
4	probaC_1.0_0.8_0.0_0.0	17944
5	probaC_1.0_0.6_0.0_0.0	16922
6	probaC_1.0_0.4_0.0_0.0	15759
7	probaC_1.0_0.2_0.0_0.0	14921
8	probaC_1.0_0.0_0.0_0.0	14147
9	mem2	14109
10	spiteful	14073
11	probaC_1.0_0.0_0.0_0.2	6497
12	probaC_1.0_0.2_0.0_0.2	6080
13	probaC_1.0_0.0_0.2_0.0	4476
14	probaC_1.0_0.4_0.0_0.2	4410
15	winner12	4343
16	probaC_1.0_0.2_0.2_0.0	4254
17	probaC_1.0_0.4_0.2_0.0	3545
18	hard_tft	3366
19	soft_majo	2603
20	probaC_1.0_0.0_0.0_0.4	2544

Note that the eighth corresponds to the spiteful probabilistic version. Both spiteful are not side by side because of statistical fluctuations.

In order to test the robustness of the strategies identified, and in particular to see what they give when they are among a few probabilistic strategies, we have composed a set with the first 20 of `ProbaCD_K=5`, the 1,024 of the `Mem(1,2)` and those of `Select`. We obtain:

rank	strategy identification	population
1	probaC_1.0_0.2_0.0_0.2	5255
2	tft_spiteful	4945
3	probaC_1.0_0.4_0.0_0.2	4877
4	probaC_1.0_0.2_0.0_0.4	4415
5	winner12	4331
6	mem12_ccCDCDDCDD	4331
7	probaC_1.0_0.6_0.0_0.2	4081
8	mem12_ccCDCDDDDD	3557
9	spiteful_cc	3557
10	probaC_1.0_0.8_0.0_0.0	3551
11	mem12_ccCCCDDDDD	2766
12	probaC_1.0_0.0_0.0_0.2	2614
13	probaC_1.0_0.6_0.0_0.0	2497
14	gradual	2469
15	mem12_ccCDDDDCDD	2368
16	probaC_1.0_0.0_0.0_0.4	2313
17	mem12_ccCCCDDCDD	2275
18	mem12_ccCCDDDCDD	2233
19	probaC_1.0_0.4_0.0_0.0	2059
20	mem2	1864

The sixth in the ranking is `winner12`, the eighth in the ranking is `spiteful_CC`. The eleventh begins with `cc` and becomes angry when the other defects twice in succession.

The results (and many others that confirm them) are very clear: the best known strategies are always well ranked (although they come from experiments and selection processes where only deterministic strategies are involved). Reciprocally, the new probabilistic strategies identified succeed very well in envi-

ronments almost exclusively composed with deterministic strategies (as in the last experiment).

5. Conclusion

In-depth and systematic experimentation in a general evolutionary model, parametrised with probabilistic strategies using a one-round memory, leads to stable results. An additional analysis allows us to identify a parameter able to anticipate the efficiency of a strategy. The double condition that we have extracted and which seems to have never been noticed leads us to define a new class of strategies (gentle spiteful).

In order to anticipate the success of a probabilistic strategy with a memory-one strategy, we have experimentally see that the double criterion `p1=1` and `p'=p2+p3+p4` as small as possible is very efficient. An interpretation is that to win an evolutionary competition or only to succeed properly, it is necessary to survive when the widespread cooperation is established. It is then necessary to cooperate with the cooperating strategies, hence the `p1=1`. The non null choice for each parameter `p2 p3 p4` is also meaningful. We show here that the best results are obtained considering `p* = 0.266 p2 + 0.138 p3 + 0.277 p4`

The members of this family are systematically at the top of all the rankings of evolutionary competitions that one can imagine. This is also true when one changes the initial set of strategies to introduce many extortioner strategies, equalizer strategies or ZD strategies. Moreover, the strategies identified in accordance with `p'` are robust and efficient in sets composed in a variety of ways, for example containing only deterministic strategies. As a result, they join the family of the best-known strategies listed in [14].

6. Appendix

List of the 21 strategies constituting `Select`.

1. `all_c`: I always cooperate.

2. `all_d`: I always defect.
3. `tit_for_tat`: I cooperate at first, then at the nth round I play what my opponent played at round $n-1$.
4. `spiteful`: I cooperate at first and as long as my opponent cooperates, but as soon as it defects I defect indefinitely.
5. `soft_majo`: I cooperate at first and as long as my opponent has cooperated more or as much as it defected in the past; otherwise I defect.
6. `hard_majo`: I defect at first and as long as my opponent has defected more or as much as it cooperated in the past; Otherwise I cooperate.
7. `per_ddc`: I play periodically `d, d, c, d, d, c, ...`
8. `per_ccd`: I play periodically `c, c, d, c, c, d, ...`
9. `mistrust`: I defect at first, then I play at the nth round what my opponent played at round $n-1$.
10. `per_cd`: I play periodically `c, d, c, d, c, d, ...`
11. `pavlov`: I cooperate at first, then I always cooperate, except when it and I did not play the same thing in the previous round.
12. `tf2t`: I cooperate in the two first rounds, then I always cooperate at the nth round, unless my opponent has defected during the rounds $n-1$ and $n-2$.
13. `hard_tft`: I cooperate in the two first rounds, then I always cooperate in the nth round, unless my opponent has defected in round $n-1$ or in round $n-2$.
14. `slow_tft`: I cooperate in the first two rounds, then I would defect when my opponent defects twice in succession, and I will not cooperate once my opponent has cooperated twice in succession.
15. `gradual`: I cooperate in the first round and when the following rule is not applied: every time my opponent betrays me, I count the number n of its past defections and I defect n times consecutively followed by two cooperations.
16. `prober`: I play defect-cooperate-cooperate (`d c d`) for the first three rounds;

Then,if my opponent has not defected in rounds 2 and 3, I always defect; Otherwise I play `tit_for_tat`.

17. `mem2`: I start by playing two rounds like `tit_for_tat`; Then I change my behavior for two rounds depending on the results of the last two rounds, using the following rules: (A) If the last two rounds were `[c c] [c c]`, I play `tit_for_tat` ; (B) If the last round was `[c d]` or `[d c]` I play `hard_tft`; (C) In all the other cases I play `all_d`. Moreover if, at any moment, my opponent defects twice in succession, I play definitively `all_d` [27].
18. `winner12` (the winner of the `Mem(1,2)` set); I cooperate for the first two rounds and then I play using the table: `[c cc]->c [c cd]->d [c dc]->c [c dd]->d [d cc]->d [d cd]->c [d dc]->d [d dd]->d`
19. `winner21` (the winner of the `Mem(2,1)` set); For the first two rounds I play `d c` then I play using the table: `[cc c]-> c [cc d]->d [cd c]->c [cd d]->d [dc c]->c [dc d]->d [dd c]->d [dd d]->d`
20. `tft_spiteful`: I play `tit_for_tat`, except if my opponent defects twice in succession, then I begin to defect indefinitely.
21. `spiteful_cc`: I cooperate for the first two rounds then I play `spiteful`.

7. References

[1] W. H. Press, F. J. Dyson, Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent, Proceedings of the National Academy of Sciences 109 (26) (2012) 10409–10413.

[2] C. Adami, A. Hintze, Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything, Nature communications 4.

[3] C. Adami, A. Hintze, Corrigendum: Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything, Nature communications 5.

[4] C. Hilbe, M. A. Nowak, K. Sigmund, Evolution of extortion in iterated prisoner's dilemma games, Proceedings of the National Academy of Sciences 110 (17) (2013) 6913–6918.

[5] C. Hilbe, T. Röhl, M. Milinski, Extortion subdues human players but is finally punished in the prisoner's dilemma, Nature communications 5.

[6] C. Hilbe, M. A. Nowak, A. Traulsen, Adaptive dynamics of extortion and compliance, PloS one 8 (11) (2013) e77886.

[7] J. Liu, Y. Li, C. Xu, P. Hui, Evolutionary behavior of generalized zero-determinant strategies in iterated prisoner's dilemma, Physica A: Statistical Mechanics and its Applications 430 (2015) 81–92.

[8] M. Milinski, C. Hilbe, D. Semmann, R. Sommerfeld, J. Marotzke, Humans choose representatives who enforce cooperation in social dilemmas through extortion, Nature communications 7.

[9] A. Szolnoki, M. Perc, Defection and extortion as unexpected catalysts of unconditional cooperation in structured populations, Scientific reports 4.

[10] A. Szolnoki, M. Perc, Evolution of extortion in structured populations, Physical Review E 89 (2) (2014) 022804.

[11] A. J. Stewart, J. B. Plotkin, From extortion to generosity, evolution in the iterated prisoner's dilemma, Proceedings of the National Academy of Sciences 110 (38) (2013) 15348–15353.

[12] H. Dong, R. Zhi-Hai, Z. Tao, Zero-determinant strategy: An underway revolution in game theory, Chinese Physics B 23 (7) (2014) 078905.

[13] B. Beaufils, J.-P. Delahaye, P. Mathieu, Our meeting with gradual, a good strategy for the iterated prisoner's dilemma, in: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems, ALIFE V, The MIT Press/Bradford Books, 1997, pp. 202–209.

[14] P. Mathieu, J.-P. Delahaye, New winning strategies for the iterated prisoner's dilemma, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1665–1666.

[15] S. Mittal, K. Deb, Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives, IEEE Transactions on Evolutionary Computation 13 (3) (2009) 554–565.

[16] R. M. Axelrod, The evolution of cooperation, Basic books, 2006.

[17] B. Beaufils, J.-P. Delahaye, P. Mathieu, Complete classes of strategies for the classical iterated prisoner's dilemma, in: International Conference on Evolutionary Programming, EP7, Vol1447, Springer, 1998, pp. 33–41.

[18] K. Sigmund, The calculus of selfishness, Princeton University Press, 2010.

[19] B. Beaufils, P. Mathieu, Cheating is not playing: Methodological issues of computational game theory., In Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06) 141 (2006) 185–189.

[20] P. Mathieu, B. Beaufils, J.-P. Delahaye, Studies on dynamics in the classical iterated prisoner's dilemma with few strategies, in: European Conference on Artificial Evolution, Springer, 1999, pp. 177–190.

[21] G. Skiba, M. Starzec, A. Byrski, K. Rycerz, M. Kisiel-Dorohinicki, W. Turek, D. Krzywicki, T. Lenaerts, J. C. Burguillo, Flexible asynchronous simulation of iterated prisoner's dilemma based on actor model, Simulation Modelling Practice and Theory 83 (2018) 75–92.

[22] A. Rapoport, A. M. Chammah, Prisoner's dilemma: A study in conflict and cooperation, Vol. 165, University of Michigan press, 1965.

[23] G. Kendall, X. Yao, S. Y. Chong, The iterated prisoners' dilemma: 20 years on, World Scientific Publishing Co., Inc., 2007.

[24] J. Li, P. Hingston, G. Kendall, Engineering design of strategies for winning iterated prisoner's dilemma competitions, IEEE Transactions on Computational Intelligence and AI in Games 3 (4) (2011) 348–360.

[25] C. O'Riordan, et al., A forgiving strategy for the iterated prisoner's dilemma, Journal of Artificial Societies and Social Simulation 3 (4) (2000) 56–58.

[26] E. Tzafestas, Toward adaptive cooperative behavior, in: Proceedings of the Simulation of Adaptive Behavior Conference, Paris, Citeseer, 2000.

[27] J. Li, G. Kendall, The effect of memory size on the evolutionary stability of strategies in iterated prisoner's dilemma, IEEE Transactions on Evolutionary Computation 18 (6) (2014) 819–826.

[28] M. C. Boerlijst, M. A. Nowak, K. Sigmund, Equal pay for all prisoners, The American mathematical monthly 104 (4) (1997) 303–305.

[29] A. J. Stewart, J. B. Plotkin, Small groups and long memories promote cooperation, Scientific reports, Nature Publishing Group 6 (2016) 26889.

[30] L. Becks, M. Milinski, Extortion strategies resist disciplining when higher competitiveness is rewarded with extra gain, Nature communications 10 (1) (2019) 783.

[31] E. Akin, The iterated prisoner's dilemma: good strategies and their dynamics, Ergodic Theory, Advances in Dynamical Systems (2016) 77–107.