National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

Department of Information Technology

technical report NUIG-IT-170900

A forgiving strategy for the Iterated Prisoner's Dilemma

C. O' Riordan (NUI, Galway)

A *forgiving* strategy for the Iterated Prisoner's Dilemma

Colm O' Riordan
Department of Information Technology,
National University of Ireland, Galway.

September 17, 2000

Abstract

This paper reports results obtained with a strategy for the Iterated Prisoner's Dilemma. The paper describes a strategy that tries to incorporate a technique to forgive strategies that have defected or retaliated, in the hope of (re-)establishing cooperation. The strategy is compared to well-known strategies in the domain and results presented. The initial findings, as well as echoing past findings, provides evidence to suggest a higher degree of forgiveness can be beneficial and may result in greater rewards.

1 Introduction

This paper adds yet another strategy to the existing collection of strategies studied in the domain. The proposed strategy contains a technique which forgives defecting strategies in the hope of re-establishing cooperation. The motivations behind the design decisions are discussed with an explanation of the different features. We compare the strategy to well-known strategies and present results.

Section 2 discuss the Prisoner's Dilemma and the Iterated Prisoner's Dilemma.

Section 3 discusses well-known strategies and discusses the properties found that render strategies successful.

Section 4 introduces *forgiving*, a strategy that incorporates most of the salient features of the well-known successful strategies, but attempts to improve performance by overcoming some of the shortcomings of these strategies.

Section 5 discusses our results which compare *forgiving* to other strategies in an evolutionary environment.

The final section presents some conclusions drawn from this work and outlines some future potential work.

2 Prisoner's Dilemma and the Iterated Prisoner's Dilemma

In the prisoner's dilemma game, there are two players who are both faced with a decision—to either cooperate or defect. The decision is made by a player with no knowledge of the other player's choice. If both cooperate, they receive a specific punishment (or reward). If both defect they receive a larger punishment. However, if one defects, and one cooperates, the defecting strategy receives no punishment and the cooperator a punishment (the sucker's payoff). The game is often expressed in the canonical form in terms of pay-offs:

		Player 1	
		C	**D**
Player 2	**C**	(3,3)	(0,5)
	D	(5,0)	(1,1)

where the pairs of values represent the pay-offs for players `Player 1` and `Player 2` respectively. The prisoner's dilemma is much studied problem due to it's far-reaching applicability in many domains. The prisoner's dilemma and applications has been described in [4][5][7] (biology) [8] (economics) and [3] (politics).

The game becomes more interesting in the iterated version where 2 players will play numerous games (the exact number not known to either player). Note that some research has indicated that it is not necessary to look to the iterated versions for more interesting behaviour to occur. Work by Epstein [6] into spatial zones indicate that more interesting behaviour (e.g mutual cooperation) can emerge and exist in the non-iterated version of the game.

A computer tournament (Axelrod)[1] was organised to pit strategies against each other in a round-robin manner. The winning strategy was *tit-for-tat*; this strategy involved cooperating on first move and then mirroring opponents move on all subsequent moves.

No best strategy exists; the success of a strategy depends on the other strategies present. For example, in a collection of strategies who defect continually (*ALL-D*) the best strategy to adopt is *ALL-D*. In a collection of strategies adopting a *tit-for-tat* strategy, an *ALL-D* strategy would not perform well

3 Strategies

It is instructive to look at other approaches and classify the strategies. The list is by no means exhaustive:

periodic: strategies play C or D in a periodic manner. Common strategies: *ALL-C*, *ALL-D*, *CD**, *DC**, *CCD**, etc.

random: strategies that have some random behaviour. Totally random, or one of the other types (e.g. periodic) with a degree of randomness.

based on some history of moves: *tit-for-tat* (C initially, then D if opponent defects, C if opponent cooperates), *spiteful* (C initially, C as long as opponent cooperates, then D forever), probers (play some fixed string, example (DDC) and then decides to play *tit-for-tat* or *ALL-D* (to exploit non-retaliatory), *soft-majo* (C initially, then cooperate if opponent is not defecting more than cooperating).

There are many variations on the above type of strategies.

The initial results and analysis (which were echoed in later tournaments) showed that the following properties seemed necessary for success—niceness (cooperate first), retaliatory, forgiving and clear.

Beaufils et al [2] question that last property and develop a strategy *gradual*[1] which is far more complex than *tit-for-tat* and outperforms *tit-for-tat* in experiments.

In a second tournament [1], of the top 16 strategies, 15 were found to be nice. These results seem to indicate that cooperative strategies are useful if there is a high chance the strategies will meet again.

4 The *forgiving* strategy

The benefits of initial cooperation, retaliation and forgiveness are clear from the initial experiments. If we look at *tit-for-tat*, we can identify techniques to overcome it's shortcomings. There are certain aspects where new heuristics (with parallels in human interactions in the real world) that could be used to improve upon 'weak' parts of *tit-for-tat's* performance.

If we consider CD* against tit-for-tat, we get:

[1] gradual performs like *tit-for-tat*, in that it cooperates on the first move. It retaliates upon defection. On the first defection it responds with a defection, followed by 2 cooperations. Following the second defection, it responds with 2 Ds, followed by 2Cs and so forth.

```
tit-for-tat   : C C D C D C D C D C
per-cd        : C D C D C D C D C D
```

The per-cd repeatedly plays a C followed by a D irrespective of tit-for-tat's move. This results in the two strategies alternating between receiving the maximum payoff and the sucker's payoff.

Tit-for-tat and other strategies could improve performance by recognising non-nice, naive strategies. Upon recognition of these strategies, a less forgiving approach can be adopted. The goal here is to be more far-sighted with respect to clearly non-nice strategies.

Another (and possibly more serious) shortcoming of *tit-for-tat* is the prevalence of spiraling into ongoing retaliation.

Consider *tit-for-tat* playing against *nasty-tit-for-tat* (tit-for-tat but attempts to exploit non-retaliatory strategies by playing a DD with some probability. (Note this can also happen with tit-for-tat in noisy environments)

```
tit-for-tat         : C C C C C C C D D D...
nasty-tit-for-tat   : C C C C C C D D D D...
```

Once *nasty-tit-for-tat* plays 2 Ds, the two strategies are locked in a spiral of mutual defections. This will not be broken until one of the two strategies plays two Cs.

We can also improve upon tit-for-tat by attempting to identify spirals and trying to re-establish cooperation via playing CC. This represents a type of forgiveness not present in other strategies. A CC is played following a set of mutual Ds. It is also important to limit the amount of times an effort should be made to re-establish cooperation.

Our strategy attempts to take these two factors into account

- Don't be exploited by periodic strategies
- Try to re-establish cooperation by forgiving

Note that the above modifications do not violate the first three recommendations (generally accepted) forwarded by Axelrod—never defect first, be retaliatory, be forgiving.

There are cases when the exploitation of periodic strategies can damage performance: where a pattern is recognised as a periodic strategy and we adopt an ALL-D approach to avoid exploitation. This can quickly result in a spiral of mutual defections if the opponent is not really periodic (but appears to be).

The degree of forgiveness in our strategy is of length 2, i.e we play two consecutive Cs. The length of the DD strings is set to 5. (i.e once 5 pairs of defections are encountered an effort is made to re-establish cooperation).

In summary, our strategy is tit-for-tat like, with the following amendments—exploit periodic strategies and forgive when interactions are spiraling into an ongoing defections.

4.1 Results - tournament

The initial experiments carried out with the strategy was in a round-robin tournament with 37 other well-known strategies[2].

The initial results, Table 1, showed that the strategy performed well, reaching a position of second. The only strategy that outperformed it was the *gradual* strategy. We see, however, in the next section that in an environmental setting the *forgiving* strategy performs better.

```
Top ten results:
1             gradual = 106277
2           forgiving = 104086
3        soft_spiteful = 102553
```

[2]The strategies chosen were those included as default in the simulation package created by Beaufils and Delahaye. Available at http://www.lifl.fr/IPD/ipd.frame.html

4	soft_joss =	102487
5	c_then_per_dc =	102345
6	hard_prober =	102341
7	tit_for_tat =	100319
8	doubler =	99409
9	prober4 =	97761
10	worse_and_worse3 =	95484

Table 1

4.2 Results - evolution

In the evolutionary simulation, each successive generation contains strategies with frequency proportional to their score in the current generation.

Again, 38 strategies were included initially. The performance of the strategies over a number of generations was plotted (Figure 1). We also include the top ten results following the 1^{st} and 21^{st}.

Strategy	Proportion in next	generation
1	gradual =	119
2	forgiving =	116
3	soft_spiteful =	115
4	soft_joss =	114
5	c_then_per_dc =	114
6	hard_prober =	114
7	tit_for_tat =	112
8	doubler =	111
9	prober4 =	109
10	worse_and_worse3 =	107

Table 2

Strategy	Proportion in next	generation
1	forgiving =	449
2	gradual =	419
3	soft_joss =	393
4	soft_spiteful =	388
5	tit_for_tat =	335
6	doubler =	309
7	c_then_per_dc =	263
8	soft_tf2t =	206
9	hard_prober =	155
10	worse_and_worse3 =	143

Table 3

This trend continues as illustrated in Figure 1, with *forgiving*'s representation in the population greater than that of any of the others.

Following these initial results we wished to investigate which of the two aspects of the *forgiving* strategy accounted for its good performance (its exploitation of periodic strategies or its ability to re-establish cooperation).

The performance of the strategy in the environmental setting indicates that its exploitation of periodic strategies, while useful, is not necessary for its success as the strategies (periodic) upon which it preys die off at a relatively early stage (e.g., *ALL-D*, *per-ddc*).

To provide empirical evidence, we also include two variations—*forgiving-1* which does not exploit periodic strategies but attempts to re-establish cooperation and *forgiving-2* which attempts to exploit periodic strategies only. The graph in Figure 2 shows their performance.

As can be seen, the two strategies that attempt to forgive and re-establish cooperation flourish.

5 Conclusion

This paper presents another strategy for the Iterated Prisoner's dilemma. The strategy outperforms other well-known strategies by introducing two new ideas to the strategy. The first extension is that of attempting to prevent exploitation by naive periodic strategies. This is found to be a useful addition.

A more useful extension is that of breaking spirals of defection by re-establishing cooperation by playing a *CC*. This provided a greater advantage in the

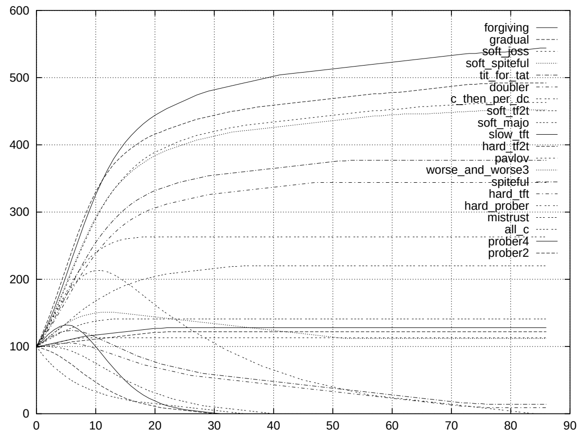

Figure 1: *forgiving* in an Ecological simulation for the Iterated Prisoner's Dilemma

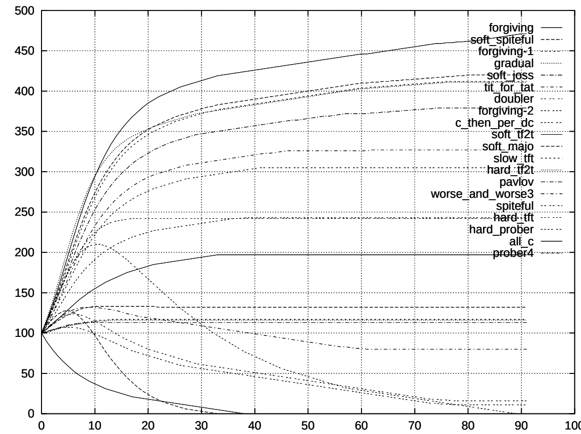

Figure 2: *forgiving* variants in an Ecological simulation for the Iterated Prisoner's Dilemma

experiments and we conclude that it is this feature that gives the advantage to this strategy.

6 Further Work

Ongoing and proposed work include further experimentation with length of strings of Cs to be used to re-establish cooperation. The number of times a strategy should attempt to re-establish cooperation is also under investigation. Evolutionary computing techniques are being used to breed strategies to help identify optimal values for this variable in different settings.

Another strand of work stemming from this study of *forgiving*, is the study of strategies working in an environment with a certain degree of noise. In this setting, there is a higher potential for mutual defection. Hence, it seems plausible that the type of forgiveness described in this paper may be of greater benefit.

References

[1] R. Axelrod. *The Evolution of Cooperation.* Basic Books, New York, 1984.

[2] B. Beaufils, J. P. Delahaye, and P. Mathieu. Our meeting with gradual, A good strategy for the iterated prisoner's dilemma. In C. Langton, editor, *Artificial Life, Proc. 5th International Workshop on the Synthesis and Simulation of Living Systems*, volume 5, 1996. Artificial Life 5, Nara, Japan, May 16-18.

[3] Jonathon Bendor and Dilip Mookherjee. Institutional structure and the logic of ongoing collective action. *American Political Science Review*, 81:129–154, 1987.

[4] L.A. Dugatkin. Do guppies play tit for tat during predator inspection visits? *Behavioral Ecology and Sociobiology*, 23:395–399, 1988.

[5] L.A. Dugatkin. N-person games and the evolution of co-operation: A model based on predator inspection in fish. *Journal of Theoretical Biology*, 142:123–135, 1990.

[6] Joshua M. Epstein. Zones of cooperation in demographic prisoner's dilemma. Technical report, Santa Fe Institute, 1997.

[7] Michael Lombardo. Tree swallows and tit for tat: Response to koenig. *Ethology and Sociobiology*, 11:521–528, 1990.

[8] Elliot Sober. Stable cooperation in iterated prisoner's dilemmas. *Economics and Philosophy*, 8:127–139, 1992.