Métodos de Apoio à Decisão – Trabalho 3

João Pedro PEDROSO

DCC - FCUP, Maio de 2016

Consider the *prisoners' dilemma* with the following payoff matrix:

Player 1	Player 2 Cooperate (`"C"`)	Player 2 Defect (`"D"`)
Cooperate (`"C"`)	5,5	0,9
Defect (`"D"`)	9,0	1,1

Take into account the following strategies for the **repeated variant** of the game:

1. cooperate in the initial 10 moves; then, cooperate if the opponent cooperated at least once in the last 10 moves, otherwise defect;
2. defect in the initial 10 moves; then, defect if the opponent defected at least once in the last 10 moves, otherwise cooperate;
3. random: cooperate with probability 50%, defect with probability 50%;
4. *tit-for-tat* (TFT): start cooperating, then play what the opponent played in the last move.

A tournament of the repeated game

In one tournament of the repeated game, agents for each of the strategies compete in a *round robin* competition (each agent plays in turn against every other). For each game, the number of moves is not fixed: after the initial 50 moves, at each repetition there is a probability of 0.00346 for the game to finish. The score in a game for a given strategy is the sum of points obtained divided by the number of moves in that game. The winner of the tournament is the strategy with the largest score accumulated in all the games.

Evolution

In the evolutionary version of the prisoners' dilemma, an *iteration* is a tournament within all the current *players*. At the end of an iteration, each player may choose a different strategy, with a probability proportional to the success of that strategy in the previous iteration. In this case, some more strategies are known to work well (here, a *punishment* is a defection immediately after the other player has defected):

1. *generous tit-for-tat* (GTFT): part of the defections of the opponent are not "punished";
2. *contrite tit-for-tat* (CTFT): a player does not punish a defection of the opponent immediately after its own defection;
3. *pavlov*: if the previous move was "successful", keep it; otherwise, change it (in this context, to be successful means to have a high return: either having defected and the opponent cooperated, or having both players cooperated).

Questions

1. Design a strategy, different of the above, for a playing tournament. Test it by simulating the tournament, and report the performance obtained for each of the strategies.
2. Do the same for the evolutionary game. Use 1000 iterations, or more if you find it relevant for your analysis. Notice that most likely the game will have periods with very different "dominant populations"; visualizing their evolution is important for the analysis.
3. Prepare your strategy to play in a tournament with the other groups, by implementing it as a Python function `play`, as follows:

```
def play(own,oth):
    if len(oth) == 0 or oth[-1] == "C":
        return "C"
    else:
        return "D"
```

(The strategy implemented is tit-for-tat.) Your filename should be called
`pXXXXXXXXX[pYYYYYYYYY].py`,
where `Xs` and `Ys` specify the student numbers.

Describe all the assumptions you have considered.

Note: *the deadline for handing the report is 19/MAY, with presentation in the following week. Your report should ideally have 4 pages. Each working group should hand a printed version and send an e-mail with the report and programs. For ecological reasons, please to not bind your report; just insert it in a transparent file folder.*