COMP3160 ARTIFICIAL INTELLIGENCE

Assignment 2 FINAL (Weight: 20%)

Evolutionary Algorithms for Adversarial Game Playing

Due: 11:55pm, Nov 06, 2020 (Friday, Week 13)

The goal of this assignment is to appreciate the efficacy of Evolutionary Algorithms, specifically Genetic Algorithm (GA), in the context of game theory. In this assignment you will be using the DEAP package for Genetic Algorithm in order to evolve strategies for repeatedly playing **3-Person Prisoners Dilemma** described below with a different storyline:[1]

> Three old friends, P_1, P_2 and P_3, always wanted to skydive but never got the opportunity. An opportunity arose when they met each other recently in Sydney for a class reunion, and decided to go for it. There are two skydiving packages available: the *BasicTandem* package (pay a certain fee and dive with an instructor), and the *TandemPlus* package (pay an additional \$600, and also get specially videoed from multiple angles while diving, and keep the video). All three friends think *TandemPlus* is better than *BasicTandem*. They also think it is not worth the extra \$600, but would be willing to pay up to \$300 extra for it instead. It is understood that they will evenly split the total cost of their skydiving. Nonetheless they are thrilled at the prospect of this joint adventure, and the ensuing pleasure is valued at \$400 by each. Each friend orders the package she wants without consultation or communication with others. If you were P_1, would you o *lus* package)?

	B	0	2	
P_i	**P**	1	3	

Table 1: *Payoffs to P_i, under how many of* j k $t = \$100$.

> The payoff matrix for *3PD* is given in Table 1. To see how this table is meant to be used, suppose both P_1 and P_2 choose the *basic* package, but P_3 chooses the *plus* package. We want to calculate P_2's payoff. Setting $i = 2$ and $\{j, k\} = \{1, 3\}$, we see P_i plays **B** (so payoffs in top row), and **only one** of P_j and P_k plays **B** (so we restrict to the column under **1B**); and determine that P_2's payoff is 2 (valued at \$200). It is easily verified that that is the case. The extra cost \$600 due to P_3's choice of **P** is evenly split between the three, P_2 contributing \$200 towards it. From the perspective of P_2, this extra cost is more than compensated by the pleasure she gets (valued \$400), and so her net gain is \$200, represented as 2 in Table 1. The payoffs to each of the three friends under alternative arrangements (e.g., if all of them opt **P**) can be similarly verified.

You will be using the DEAP package for Genetic Algorithm in order to evolve strategies for playing *Iterated 3-Player Prisoners Dilemma* (3IPD). Two papers on the application of GA to PD – one to IPD and the other to nIPD – are provided in the Assignment folder.

[1] Also called the *Unscrupulous Diners Dilemma*

Task Specification

Note: You are advised to go through the two supplied papers:

i "Using GA to Develop Strategies for IPD," by A Haider, and

ii "An Experimental Study of N-Person IPD Games," by X Yao and PJ. Darwen

in the given order before proceeding with the assignment tasks. Give particular attention to Sections 4.1 and 2.1 of the respective works.

1. BACKGROUND KNOWLEDGE ASSESSMENT [3 marks]

   (a) Analysing the Payoff matrix provided in Table 1, determine if a Nash Equilibrium exists for the game *3PD*. If so, identify at least one of its Nash Equilibria, and explain why it is so.

   (b) Suppose we want to represent strategies for playing 3IPD of memory depth 2 in the context of GA. How many bits shall we need to represent the individuals, and why? Answer in no more than 50 words.

   (c) Consider a strategy (individual/chromosome) of memory-depth 2 for playing 3IPD. Explain how you would represent the memory bits and the default moves in this individual.

2. IMPLEMENTATION IN PYTHON [15 marks]

   (a) Impleme

   ```
   payo                                    :
   returns payoff to individual1
   ```


   Note: payoff is determined by latest mov appropriate memory locations of the individuals and t r the game game. (Assume that the game is 3PD and memory-depth is 2.)

   (b) Implement the function:

   ```
   move_by_ind1(individual1, individual2,
                          individual3, round):
   returns individual1's move
   ```

   Note: `individual1`'s next move is based on all the three individuals' earlier moves and `individual1`'s strategy (encoded in `individual1`'s chromosome). The move to be returned can be `B/P`, or `C/D`, or `0/1` depending on your representation. Note that in early rounds some default moves are made. Assume memory-depth of 2.

   (c) Implement the function:

   ```
   process_move(individual, move, memory_depth):
   returns success / failure
   ```

   Note: `individual`'s relevant memory bits are appropriately updated based on its latest move `move` and `memory depth`.

(d) Implement the function:

```
def eval_function(individual1, individual2,
                  individual3, m_depth, n_rounds):
returns score to individual1
```

Note: `individual1` iteratively plays *3PD* against the other two for a **number of rounds** given by `n_rounds`, and its score is returned.

(e) Implement, using the DEAP package, genetic evolution of strategies for playing 3IPD. Assume a memory depth of 2. Based on your implementation, briefly describe the best 3IPD-individual you generated via GA, and what parameters (fitness function, type of crossover, mutation rate, etc.) you used for that purpose. Explain why you chose those specific parameters.

3. ANALYSIS [2 marks]

(a) Describe in English the behaviour of the 3IPD-strategy you obtained via task 2e above. Exploit any pattern you notice in it for this purpose.

(b) What would you consider to be a good counterpart of the strategy *Tit for Tat* in 3IPD? Compare it with the best strategy you obtained via task 2e in terms of traits such as being nice, being forgiving and being provocable. Make an attempt to explain any major difference in these two strategies in terms of their payoff structures.

What to Submit

You will submit two files:

1. <yourLastname_yourFirstname>_code.py.
2. <yourLastname_yourFirstname>_report.pdf.

Your code file should include all the Python codes you wrote for this assignment.

Your report file should include all the answers (including the Python codes copied-and-pasted). It must be submitted in the pdf format. It must have as cover page the one that has been supplied (as part of the document template), duly filled and signed. Also, if relevant, note in the last section anything relevant that is worth noting.

You will submit the files in two stages. In the first stage you must submit two draft files (that you will be able to update) by 11:55pm, Friday Week 12:

(a) of the program file <yourLastname_yourFirstname>_code.py including at least the implementation of functions specified in Tasks 2a and 2b.

(b) of your report file <yourLastname_yourFirstname>_report.pdf with answers to Tasks 1a-1c. You can modify these files while preparing your final version. However, failure to submit the two draft files by the required date will attract a penalty of 4 marks. The final version of these three files must be submitted by 11:55pm, Friday Week 13.