Evolutionary Algorithms for Adversarial Game Playing

The goal of this project is to appreciate the efficacy of Evolutionary Algorithms, specifically Genetic Algorithm (GA), in the context of game theory. I used the DEAP package for Genetic Algorithm in order to evolve strategies for repeatedly playing variants of *The Tragedy of the Commons* (TC). Basically, there is a group of farmers (herdsman) in a village, and there is a community grazing area ("Commons"). If the farmers limit the number of cows that each of them is allowed to graze in the Commons, everyone prospers. If a few farmers flout the convention, they benefit at some cost to others. But if too many farmers flout the convention, then everyone suffers due to over-grazing. This game is quite pertinent to the sustainability issue.

The Tragedy of the Commons is originally described by Hardin as follows(this file can be found in the 'References' folder):

As a rationale being, each herdsman seeks to maximize his gain. Explicitly or implic- itly, more or less consciously, he asks, "what is the utility to me of adding one more animal to my herd?" This utility has a negative and a positive component.

1. The positive component is a function of the increment of one animal. Since the herdsman receives all the proceeds from the sale of the additional animal, the positive utility is nearly +1.
2. The negative component is a function of the additional overgrazing created by one more animal. Since, however, the effects of overgrazing are shared by all [. . .], the negative utility for any particular decision-making herdsman is only a fraction of 1.

Adding together the component particular utilities, the rational herdsman concludes that the only sensible course for him to pursue is to add another animal to his herd. And another; and another... But this is the conclusion reached by each and every rational herdsman sharing a commons. Therein is the tragedy. Each man is locked into a system that compels him to increase his herd without limit – in a world that is limited. Ruin is the destination towards which all men rush, each pursuing his own best interest in a society that believes in the freedom of the commons. Freedom in a commons brings ruin to all.

I will look at two very simplistic models of this imagined scenario. Both of them are two-player games.

In the first model (TC_1), I assume a small grazing common land that can comfortably support two cows altogether, and each cow brings 5 units of utility to its owner. If one more cow is allowed to graze, the cows are not adequately fed, and the benefit from each cow reduces by 1. If instead two more cows are allowed in, the feed is fully inadequate for every cow because of overgrazing, and they bring zero benefit each to their owners. It is assumed that there is a tacit understanding between the two farmers that they will *cooperate*, that is, will allow in only one cow each. The payoff matrix for this model is given as follows:

Table 1: **The Tragedy of the Commons: TC_1**

		ROW	
		C	*D*
COLUMN	*C*	(5, 5)	(4, 8)
	D	(8, 4)	(0, 0)

Payoffs to: (Column, Row)

The second model (TC2) changes the narrative to some extent. The local council decides that the two farmers can take a lease of part of the "commons" for a nominal fee, and use that part exclusively for grazing their own cows. However there is a constraint: each can take an independent decision of whether to lease $\frac{1}{3}$ of the commons, or $\frac{1}{4}$. Either way, part of the commons will be left un-leased, and they will have equal grazing right over that un-leased part. Here assume that the total commons has an area of 24, and each unit area provides one unit of benefit. Again there is the tacit understanding that the two farmers will *cooperate*, that is each will take out a lease on $\frac{1}{4}$ of the commons. However there is incentive to defect, as the payoff matrix shows below. For instance, if one farmer cooperates, and the other defects, the cooperator receives $\frac{24}{4} + \frac{24-\left(\frac{24}{4}+\frac{24}{3}\right)}{2} = 11$ units of benefit, while the defector will receive $\frac{24}{3} + \frac{24-\left(\frac{24}{4}+\frac{24}{3}\right)}{2} = 13$ units.

Table 2: **The Tragedy of the Commons: TC_2**

		ROW	
		C	*D*
COLUMN	*C*	(12, 12)	(11, 13)
	D	(13, 11)	(12, 12)

Payoffs to: (Column, Row)

Specification

The main issue is the representation of a strategy for playing a game like TC_1 or TC_2 as a bit string. Represent players/strategies in following representation:

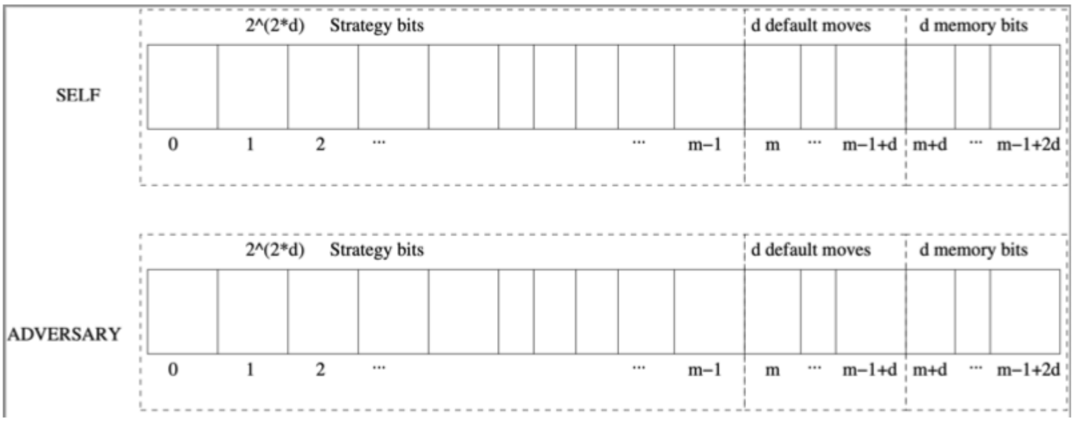

Figure 1: Representation of two players. With d as the memory depth, $m = 2^{2d}$. Total length of a player/chromosome: $2^{2d} + 2d$.

1. BackGround Knowledge assessment

This part is the insight of the project which includes basic questions and answers.

(a) Consider the *Cryptocurrency Mining* from [this New Yorker article](#) about the environmental impact of Bitcoin mining. Which of the two models of the Tragedy of the Commons, TC1 and TC2, is closer in spirit to this situation?

The shared resource in cryptocurrency mining is electric energy. This resource is not infinite, but it is still can be sufficiently extracted in many ways. Therefore, when both players are defective, a strong dominant strategy equilibrium can be obtained as in TC2 rather than get nothing as in TC1.

(b) Find all *Dominant Strategy Equilibria*, *Nash Equilibria* and *Pareto Optimal Strategy Profiles* for the game TC_1.

TC1

Dominant Strategy Equilibrium:

For Column, D is better if Row plays C but not if Row plays D. Similarly, for Row, D is strictly better if Column plays C but not if Column plays D. Thus, none of the farmer has a strong/ weak dominant strategy in TC1.

Nash Equilibrium:

A strategy profile is in Nash equilibrium if no player has any incentive to deviate from it if other players don't.

A strategy profile (C, D) is a Nash equilibrium since the payoff for row in (C, C) is no better than (C, D), and the payoff for column in (D, D) is no better than (C, D) Therefore, (4,8) is a Nash Equilibrium.

A strategy profile (D, C) is a Nash equilibrium since the payoff for column in (C, C) is no better than (D, C), and the payoff for row in (D, D) is no better than (D, C). Therefore, (8,4) is a Nash Equilibrium.

Pareto Optimal Strategy Profiles:

A Pareto optimal strategy profile is such that no player's payoff can improve without some player getting disadvantaged.

(C, C) is Pareto Optimal, since (C, C)'s farmers can not be improved to (C, D) or (D, C) without the other farmer getting disadvantaged. Furthermore, nobody in (C, C) is worse off than (D, D).

(C, D) is Pareto Optimal, since (C, D)'s farmers can not be improved to (C, C) or (D, C) without the other farmer getting disadvantaged. Furthermore, nobody in (C, D) is worse off than (D, D).

(D,C) is Pareto Optimal, since (D,C)'s farmers can not be improved to (C, C) or (C, D) without the other farmer getting disadvantaged. Furthermore, nobody in (D, C) is worse off than (D, D).

(D, D) is not Pareto Optimal, since (C,C), (D,C), (C,D) have the better values.

(c) Find all *Dominant Strategy Equilibria, Nash Equilibria* and *Pareto Optimal Strategy Profiles* for the game TC_2.

TC2

Dominant Strategy Equilibrium:

For Column, D is strictly better when Row plays C or D. Similarly, for Row, D is strictly better when Column plays C or D. Thus, TC2 has strong dominant strategy equilibrium(D,D).

Nash Equilibrium:

A strategy profile is in Nash equilibrium if no player has any incentive to deviate from it if other players don't.

A strategy profile (D, D) is a Nash equilibrium since the payoff for row in (D, C) is no better than (D, D), and the payoff for column in (C, D) is no better than (D, D) Therefore, (12, 12) is a Nash Equilibrium.

Pareto Optimal Strategy Profiles:

(C, C) is Pareto Optimal, since (C, C)'s farmers can not be improved to (C, D) or (D, C) without the other farmer getting disadvantaged. Furthermore, (C, C) has same values with (D, D).

(D, D) is Pareto Optimal, since (D, D)'s farmers can not be improved to (C, D) or (D, C) without the other farmer getting disadvantaged. Furthermore, (D, D) has same values with (C, C).

(C, D) is Pareto Optimal, since (C, D)'s farmers can not be improved to (C, C), (D, D) or (D, C) without the other farmer getting disadvantaged.

(D, C) is Pareto Optimal, since (D, C)'s farmers can not be improved to (C, C), (D, D) or (C, D) without the other farmer getting disadvantaged.

(d) Analyse the results I obtained in items 1b and 1c above. Describe any interesting difference I noted between them.

TC2 has a Strong Dominant Strategy at (D, D) but TC1 does not have one.

TC1 has its Nash Equilibriums on the diagonal but TC2 has Nash Equilibrium only at the same point with Strong Dominant Equilibrium(D, D).

TC1 has one strategy that wasn't Pareto Optimal but all of the strategies in TC2 are Pareto Optimal.

(e) **Axelrod's tournaments of Iterated Prisoner's Dilemma show that co-operation among players can emerge in a world of selfish agents. What do I expect would happen if different strategies were to play Iterated TC_1 (henceforth (ITC_1) instead? What would I expect in the case of Iterated TC_2 (henceforth (ITC_2)?**

In TC1, the payoffs when both players cooperate and defect are very different (5,5), (0,0). Therefore, because of the risk of (D, D), a properly cooperating strategy in ITC1 can succeed. However, in TC2, since (C, C) and (D, D) is the same as (12,12), the defect strategy is unconditionally advantageous in ITC2.

2. IMPLEMENTATION IN PYTHON

This part is for the explanation of how that I implement ideas into the python code.

The real code is in the python file named 'PythonCodeForProject'.

(a) **Implement the function:**
payoff_to_player1(player1, player2, game):
returns payoff

Player2 is the adversary, and payoff is determined by latest moves obtained from respective appropriate memory locations and the payoff matrix for the relevant game (TC_1 or TC_2). Assume memory-depth of 2.

(b) **Implement the function:**
next_move(player1, player2, round):
returns player1's move

player1's next move is based on both the players' earlier moves and player1's strategy (encoded in player1's chromosome). The move to be returned can be C/D or 0/1 depending on representation. In early rounds some default moves are made. Assume memory-depth of 2.

(c) **Implement the function:**
process_move(player, move, memory_depth):
returns success / failure

player’s relevant memory bits are updated based on its latest move move and its memory depth.

(d) Implement the function:
score(player1, player2, m_depth, n_rounds, game):
returns score to player1

Player1 iteratively plays the game ‘game’ against player2 for ‘n rounds’ number of rounds, and its score is returned.

(e) Using the DEAP package for genetic evolution of strategies. Assume a memory depth of 2.

i. Implement genetic evolution of strategies for playing ITC1.
ii. Modify the code I wrote as part of 2(e)i above so as to genetically evolve strategies for playing ITC2.