Assignment 2

Evolutionary Game Theory

Samuel Devine – 20463944

Part 1: Analysis and Results

Analysis of Evolved Strategies

The memory 1 algorithm always lands on the same strategy right from the first round:

Strategy Genome: [0, 1, 1, 0]

This strategy it settles on works like this:

- It starts by defecting
- If both players cooperated in the last round, it cooperates next time
- If it cooperated and the opponent didn't, it still cooperates
- If it defected and the opponent cooperated, it keeps defecting
- If both defected, it defects again

It kicks things off by defecting but if the opponent shows they're willing to cooperate, it may return the favour, but it doesn't always reward cooperation. It depends on the history. This makes it opportunistic, where it takes advantage of overly trusting opponents but is hard on those that don't cooperate.

Immediately converging to this solution is not surprising as the strategy space is so small (16 strategies). Because of this small number, the genetic algorithm easily identifies the optimal strategy for the these opponents straight away, without needing more generations to fine tune things.

Performance Against Fixed Strategies

The evolved strategy was tested in 100 round games against three different types of opponents:

- One that always cooperates
- One that always defects
- And one that plays Tit-for-Tat

Opponent	**Player Score**	**Opponent Score**
Always Cooperate	**400**	150
Always Defect	**100**	100
Tit-for-Tat	**269**	264

Insights:

- When against the Always Cooperate opponent, the strategy does take advantage by defecting at first but then cooperating after cooperating moves. The cycle is Defect -> Cooperate and this repeats every two rounds. Leaving us with payoffs of 5 -> 3 and the opponent with 0 -> 3 every two rounds leading to the final player score of 400 and opponent score of 150
- Against Always Defect, both just keep defecting, so they both finish with low scores of 100 each.

Against Tit-for-Tat

TFT is a deterministic opponent, meaning it always cooperates on the first move and mirrors what we did last round. The repeating cycle for TFT against the 0,1,1,0 strategy is:

- Defect (we get 5) -> Cooperate (we get 0) -> Cooperate (we get 3)

This cycle repeats and after 100 rounds the scores are very close with the both TFT and the player getting 8 points per cycle. The 100th round (first in the cycle) gives the player the bonus 5 points for the end score to be Player: 269 TFT: 264

Discussion of Findings

1. Immediate Convergence

The genetic algorithm lands on the best strategy straight away, in the first generation. Once that happens, there's no room for fitness improvements in future generations as it has already found the best option. This highlights an important limitation with this setup:

- Using a memory-one, binary encoded strategy, the search space is extremely small.
- Because of this, even a decently sized population (50–100) can explore the whole space right from the start, making the evolution process unneccessary. There's just not enough complexity beyond the first generation.

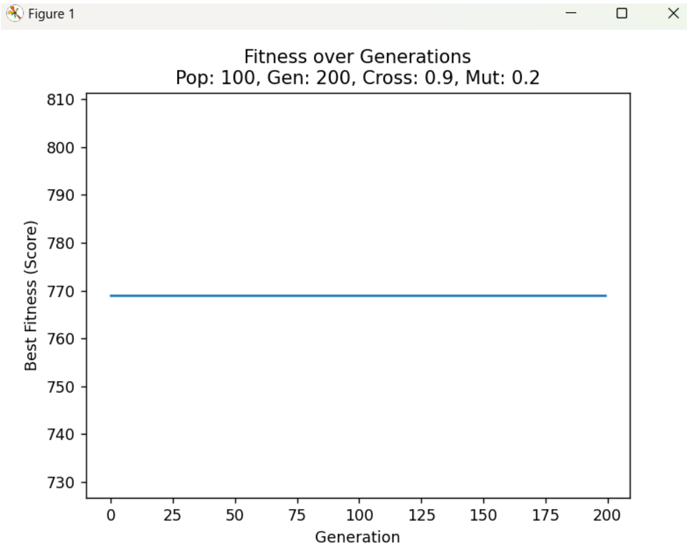
2. Strategy Effectiveness

The evolved strategy shows a good balance between exploitation and defense:

- It exploits overly cooperative opponents, getting high payoffs without being opened up to risk.
- It holds its ground against uncooperative opponents, like Always Defect. It ensures it doesn't fall behind in terms of payoff. In these cases it tends to end in both players receiving similar low scores.
- Against Tit-for-Tat, the strategy went in to a 3 stage cycle of cooperation and defection. This resulted in a fairly acceptable result. Giving 8 points to each player per cycle (5 -> 0 -> 3), not perfectly cooperative but does keep good mutual benefit.

Visualization of Fitness Progression Over Generations

Below is the fitness progression of the best individual across 200 generations:



As shown in the graph, the fitness reaches 769 by the end of the first generation and stays flat from that point onward. The plateau shows the immediate convergence of the GA on the optimal solution in the small search space. Once found, there is no further improvement across generations as the algorithm has already explored the options.

Conclusion for Part 1

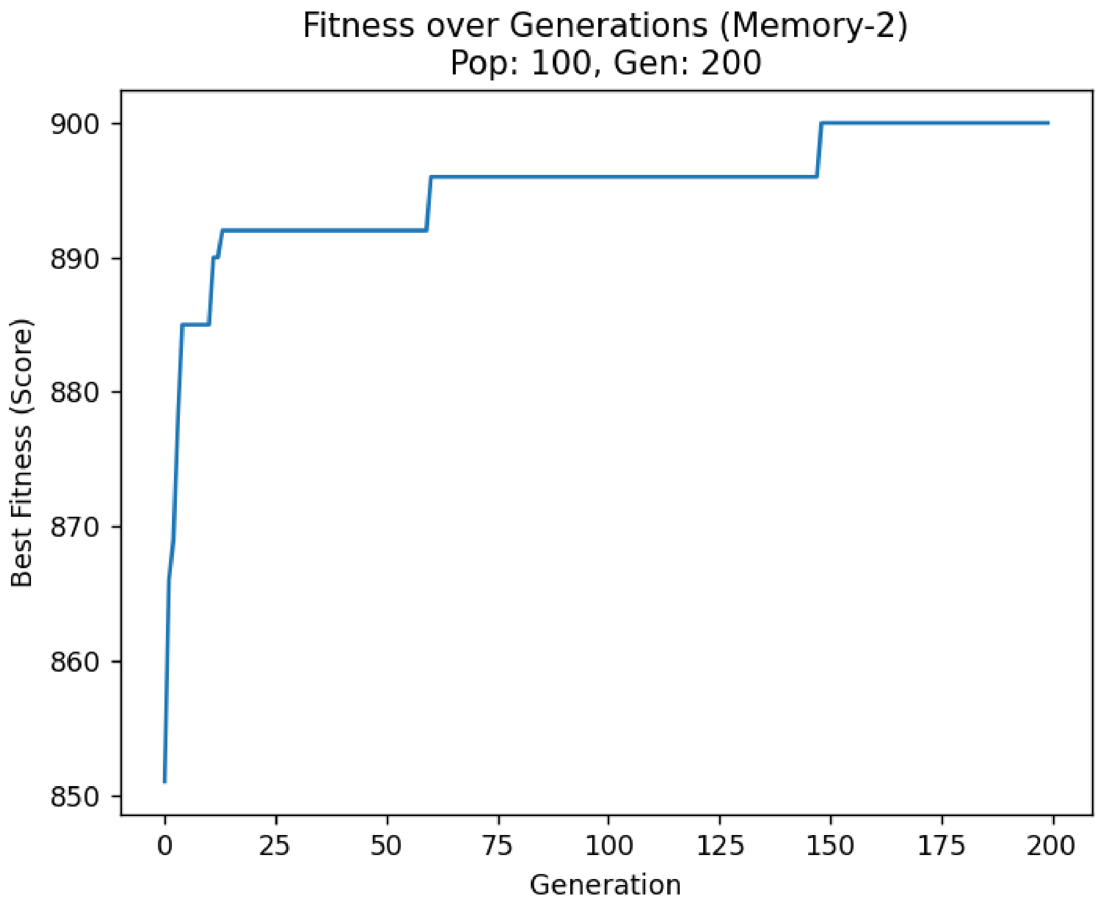
The genetic algorithm was able to evolve a strategy that performs well against a few different opponents. It does a good job balancing when to cooperate with opponents and when to defect. Strategies being limited to memory one, binary setup, the search space is very small. This means there isn't much room for evolution after the best strategy is found immediately.

Part 2 Analysis and Results

The first algorithm converged quickly to the optimal memory 1 strategy because of the small search space. To expand on this, in part two I look in to two extensions. The first was increasing the memory to memory 2, this lets more complex behaviours evolve. Then introducing noise to the TFT opponent, this is meant to add unpredictability and test the robustness of the strategies.

By extending the strategy format from memory 1 (5 genes) to memory 2 (17 genes), the strategy now looks at the last two moves from each player. Doing this gives more memory and allows strategies to recognize patterns in an opponents behaviour. The strategy space hugely increases and it lets the genetic algorithm to explore more rather than immediately converging. One of the best strategies returned was [0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1], this gave a fitness of 899. It begins by defecting then adapts as it goes on with 2-round history of player and opponent moves.

Below is a plot showing the fitness over generations for one memory 2 simulation.



The fitness improves over time unlike memory one where it plateaud immediately. This shows that the bigger complexity lets us get more gradual improvements and more evolution. This exploits 'always cooperate' and 'always defect better and gets the maximum score from both (500 and 100 respectively).

Against TFT with noise, the strategy did balance cooperation and defection well, returning an equal score (varying a bit per simulation due to noise). The noise in TFT here works by adding a 10% chance the TFT makes an error and does the opposite of what it usually would. So the strategies had to adapt to this occasional randomness for TFT. The best strategies would have to have been more forgiving after suffering from noise. This is a good adavantage of memory 2, it can recover from random defections better than memory 1. This helps keep cooperation in TFT.

Conclusion

Switching to memory 2 strategies made a big difference, it let our GA explore a lot more and gave us the chance to see some evoultion. We saw some nuanced behaviours and saw how it could adapt to noise introduced in TFT. This showed the robustness of the evolved strategies and how the best solutions had to be those which could withstand the occasional mistake and keep cooperation. It was a slightly longer computation of course but the time to find a good solution was trivially low (~5 seconds).