COSC343: Assignment 2 Report

Shivam K.C. (3746992)
May 15, 2020

1 Introduction

In a 2D grid-based game, that pits two populations of creatures (red and blue) against each other, I am tasked with implementing an agent that controls one population [1]. The game's objective is for the agent to eat strawberries, placed at random on the grid, grow in size and eat opposing creatures of smaller size. The creatures grow in size by eating strawberries or opponent's creatures. My genetic algorithm will find behaviours that lead to more of my creatures surviving at the end of the game.

2 Agent Function Model

The behaviour of my creatures are governed by the agent function. The purpose of the agent function is to produce an action vector from a percept vector utilizing the chromosome of a creature. The action vector is how the game engine interprets what the creature wants to do and the percept vector informs the creature about the presence of walls, other creatures (and their sizes), and strawberries in the 24 neighbouring cells around its current position.

I choose the following model of the agent function:

1. Once the function gets the percept vector, b (size $5 \times 5 \times 3$), it flattens the percept vector to size 75×1.
2. Then, the function dot products the percept vector, b, with chromosome, c, of size 7×75 (description below) to get the action vector, y, of size 7×1. Basically,

$$y_i = \sum_{j=0}^{74} (b_j \times c_{i,j}),$$

   where i ranges from 0 to 6 inclusive.

I choose this because it fulfills the purpose of agent function: Different percepts values lead to different actions. This way the agent reacts differently to different situations. Different chromosome leads to different actions. This way different agents can exhibit different behaviour.

3 Choice of the Chromosome

I choose the chromosome of a creature to be 7×75 vector consisting of randomly assigned integers ranging between -75 to 75 (inclusive).

I choose the chromosome's size to be of 7×75 in order to fulfill the purpose of my choice of the model of the agent function above. I have the chromosome consisting of randomly assigned integers from -75 to 75 so that the creature will eventually learn that negative values mean the creature is less likely to act certain way and positive values mean the creature is more likely to act certain way.

4 Description of My Genetic Algorithm

4.1 Fitness Function

The fitness of each creature is evaluated by summing a boolean value (whether the creature survived at the end of the game) and its total energy (obtained by using the equation $s = \lfloor \log_2 e \rfloor$ where e is the energy and s is the creature's size)

I believe this fitness function will help me create energetic and "lucky" creature in the new population. Having high energy is crucial for the creature if it is to do well in the game and contribute to its agent's winning.

4.2 Selecting Parents

I choose tournament selection method to select the fitter parents. I pick a subset of n random individuals from the population k where $n = 90\% \times k$. From the set of those n individuals, two with the highest fitness are selected as the two parents. I did this so that there is only 10% chance of missing out on the fittest parents and at the same time, there is some variety.

4.3 Crossover

I create a new individual based on the chromosomes of the selected parents. For the sake of convenience, I am talking about the flattened chromosome of the new individual of size 1×525, which is later reshaped to be our default sized chromosome of 7×75. The crossover between two parents are done the following way:

1. For the indices 0 to 300 of the chromosome of the new creature, the values are obtained from parent 1 (the fitter of the two) from its positions 0 to 300 so that the new creature is more likely to move around the grid like the fittest parent.
2. For the indices 301 to 375 of the chromosome of the new creature, the values are obtained from parent 2 from its positions 301 to 375.
3. For the indices 376 to 450 of the chromosome of the new creature, the values are obtained from parent 1 from its positions 376 to 450 so that the new creature is more likely to not miss the chance to eat strawberries and gain energy like the fittest parent.
4. For the indices 451 to 525 of the chromosome of the new creature, the values are obtained from parent 2 from its positions 451 to 525.

4.4 Mutation

I choose to introduce a mutation in the child in order to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next.

After the crossover, the mutation only happens 10% of the time. If the mutation occurs, each values in the chromosome of the child will randomly change/flipped to an integer value between -75 to 75 inclusive.

4.5 Elitism

I choose to retain the fittest individual from the old population 7 times in the next generation so that the average fitness values of the new population increase and my creatures perform well. Therefore, the new population will have only 27 new creatures through crossover (maybe some of them through mutation) and 7 old creatures through elitism (given the size of the old population is 34).

5 Results

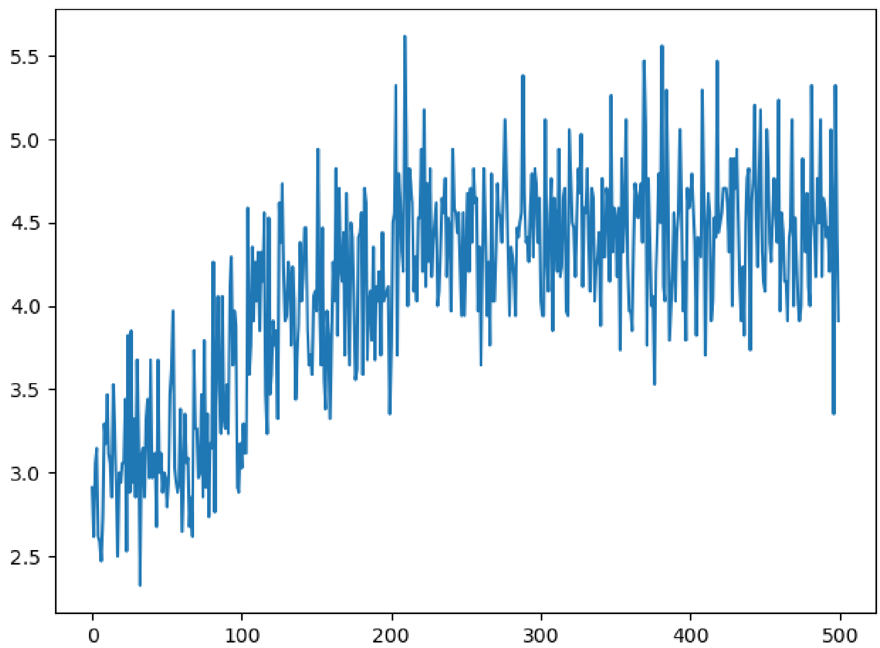

Figure 1: Graph showing how average fitness of the population changes as evolution proceeds. The x-axis is the number of generations. The y-axis is the average fitness of the population.

According to the graph above, the average fitness value of the population improves as the generation of the population increases. The average fitness of the population seems to plateau near the end. This shows that the creatures are learning to move more effectively in the 2-D grid plane.

I carry out the simulation of 500 games (my agent vs random agent) for 5 times. I find that initially at game 1, my agent always loses against the random agent. However, by game 500, my agent has learned to perform better. Thus, my agent wins 4 out of 5 times in game 500.

6 Conclusion

In conclusion, the evolution plays a big role in shaping the creature's behaviors and my agent improving as the generation increases. Particularly, the following aspects of genetic algorithm helps:

1. Initializing the chromosome with randomly assigned integers from -75 to 75 so that the creature will eventually learn that negative values mean it is less likely to act certain way and positive values mean the creature is more likely to act certain way.
2. The fitness function helps my agent create energetic and "lucky" creature in the new population. Having high energy is crucial for the creature if it is to do well in the game and contribute to the agent's winning.
3. Picking a subset of 90% of random individuals from the population during tournament selection so that there is only 10% chance of missing out on the fittest parents and at the same time, there is some variety.
4. Choosing proper crossover between the fittest parents so that the new creature is more likely to move around the grid, and not miss the chance to eat strawberries and gain energy like the fittest parent of the two.
5. Choosing to introduce a mutation (with 10% of probability) in the child in order to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next

References

[1] Szymanski, Lech. "Evolve a species in a game" 2020. PDF file.