COSC343 Assignment 2 - Snakes on a plane

Report for Shay Stevens, ID #7196262

Introduction

When given a problem it is easy to create a solution using AI based on the rules of the environment. But how do you find a solution in an unknown environment? Whenever a problem like this arises your best option is to use a genetic algorithm. GA's are able to find a solution to any problem by learning from past generations. Based on evolution a GA uses principles like elitism, mutation, parent selection and crossover to find a solution in an unknown environment. This report is about a genetic algorithm I created using a perceptron model to find the solution to a snakes vs snakes game.

Implementation and chromosome explanation

I decided to use a perceptron model. I chose this model because it is the easiest model for me to understand and it still performs well. It gets very complex very quickly so having a large field of vision and frames will make the algorithm very slow.

Let's say percepts field of vision is 3 and the number of frames is 1.

Percepts:

[x1, x2, x3,
x4, x5, x6,
x7, x8, x9]

Chromosome:

[[w1, w2, w3, w4, w5, w6, w7, w8 , w9], [w10, w11, w12, w13, w14, w15, w16, w17, w18], [w19, w20, w21, w22, w23, w24, w25, w26, w27]]

v1 = x1w1 + x2w2 + x3w3 + x4w4 + x5w5 + x6w6 + x7w7 + x8w8 + x9w9 + b

v2 = x1w10 + x2w11 + x3w12 + x4w13 + x5w14 + x6w15 + x7w16 + x8w17 + x9w18 + b

v3 = x1w19 + x2w20 + x3w21 + x4w22 + x5w23 + x6w24 + x7w25 + x8w26 + x9w27 + b
Where b = bias value.

Since the chromosome is initialized to random values v1, v2, v3 will all have different values.

If v1 is the largest the snake will go left.
If v2 is the largest the snake will go right.
If v3 is the largest the snake will go forward.

Let's say x2 is food so it's value is 2 and w20 is a large positive value all other values are zero. This will cause v3 to be the largest value and the snake will go forward into food. This is the correct action.

Now let's say x4 is an enemy snake so its value is -1. If w4 is a large negative number that makes v1 the largest value the snake will go left into an enemy snake and die. This is a very bad action which will cause the snake to have a bad fitness value so their genes aren't passed down into the next generation.

That is how the snake's chromosomes will determine its behavior.

Genetic Algorithm

Fitness Function

$$f = \mathrm{maxSize} + \frac{\mathrm{turnsAlive}}{N},$$

I am using the default fitness function for my agent. This function favors large snakes over snakes that have been alive for a long time. I have tried to alter it and make it better but the average fitness of my snakes was worse so I decided to scrap it. The Idea I had was adding a value that favors snakes that become large very quickly but it was causing my snakes to spin infinitely in a circle.

Mutation

There is a 3% chance of a mutation on any part of a snake's chromosome. I have tried multiple different values for the mutation rate and found that 3% was the best.

Elitism

My GA uses elitism so the best snakes survive into the next generation. The elitism rate is set to 20% so the best 8 snakes of the previous generation make it to the next generation.

Uniform crossover

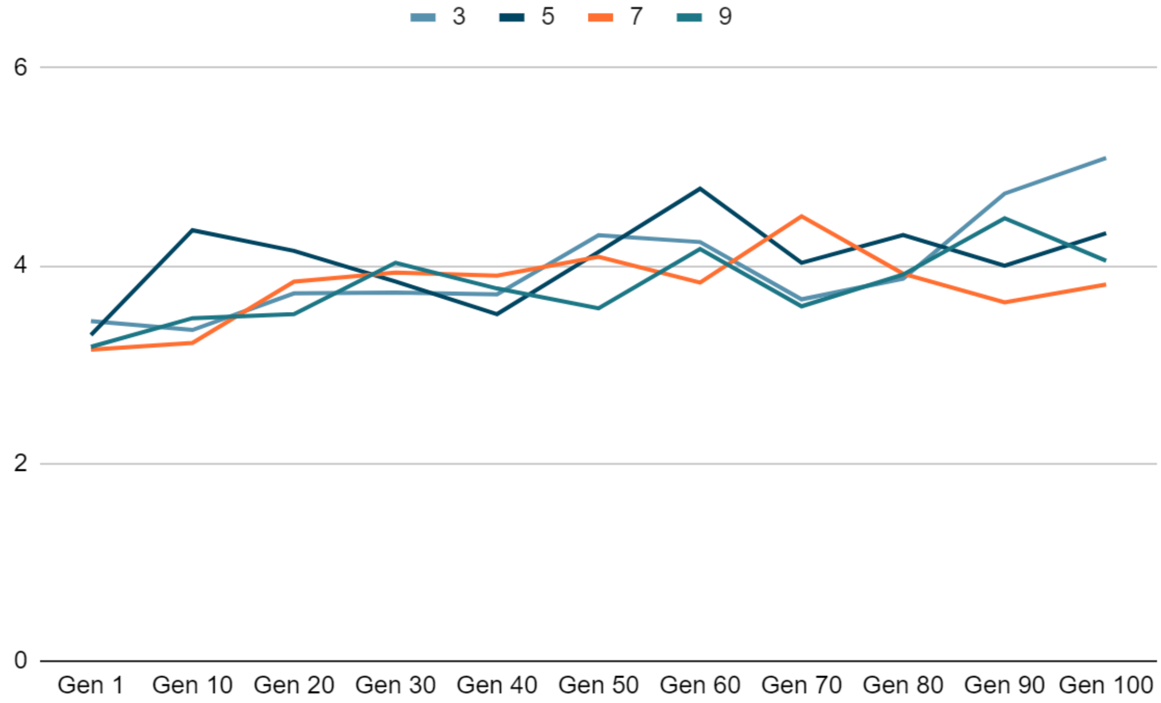
Uniform crossover is used for the parent chromosomes to create the child’s chromosome. I have also implemented single point crossover but the results were terrible. The average fitness ended up getting worse over time so I decided to scrap it.

Tournament selection

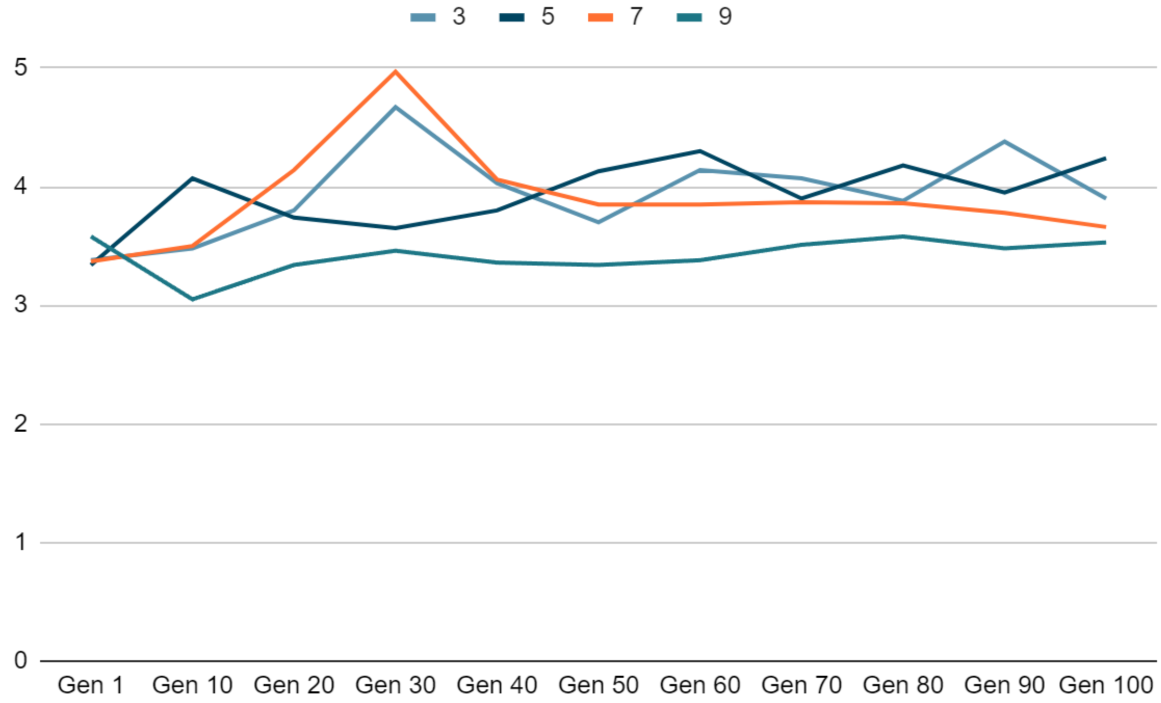
Tournament selection is used to pick the parents for the child snake. At the moment only the top two snakes in a population ever reproduce because it produces the best results. But if you want to change to a subset of the population you have the option to do so.

Evaluation:

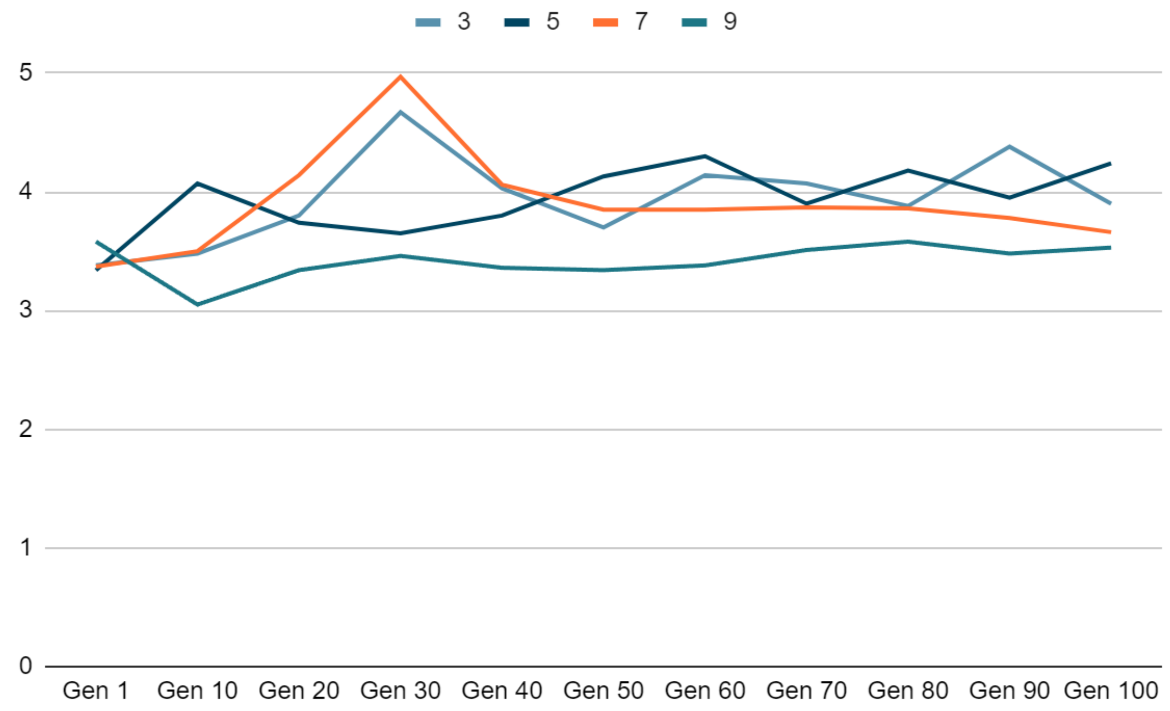
1 frame:



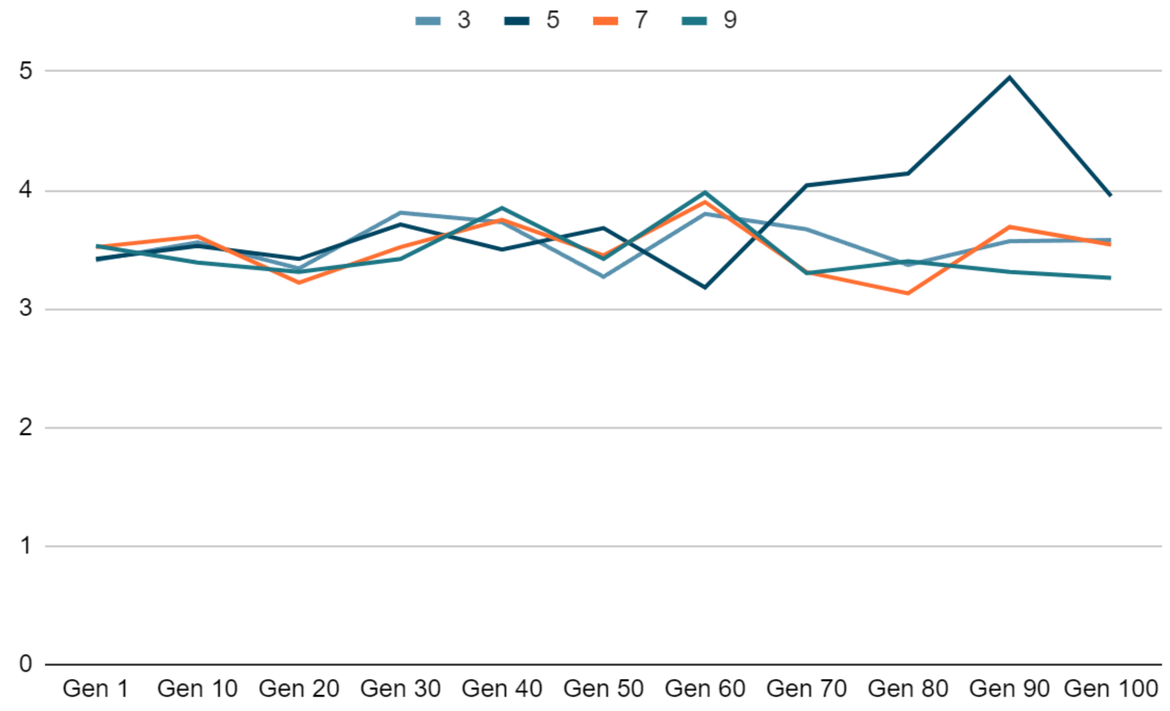
2 frames:



3 frames:



4 frames:



3, 5, 7, 9 is the percepts field of vision. The average fitness does improve over time but not consistently and with more frames it is difficult to see the average fitness improving at all. From looking at the graphs the smaller the field of vision and frames the better the agent does. This reflects with the scores as well. The highest score my agent got was 91 and that was with a field vision of 3 and one frame.

Analysis

The GA performs best with a small percept field of vision and one frame also it is the least time complex. By looking at the visualization it is pretty evident that behavior is learnt. My snakes will avoid other snakes and travel towards food. However, my snakes are hard for a random agent to beat but not unbeatable. Sometimes they will spiral into themselves. I have tested my snakes against 50 and 60 random snakes and despite this disadvantage it can still beat the random agent.

Conclusion:

My genetic algorithm is good but not the best. I feel like there are a lot of areas that need to be improved. Thinking logically with more information my agent should perform a lot better than when it has less information but this not the case I think it has to be a problem with the chromosome perhaps have more genes when there are more frames or maybe it is the fitness function and I need a better way to evaluate the fitness of the snakes.