Henry Morrison-Jones ID: 1625275

COSC343 Assignment 2 Report

Simulation

For the attributes of the simulation, I wanted to ensure that each generation occurred in a world that was simple enough to find a maxima but complex enough that their fitness would not converge to a local maxima. The number of turns in the generation was doubled to 200. This was done as a 100 turn simulation acted as a ceiling to the optimization with often over 60% of creatures surviving. This meant that Average Fitness and Highest Fitness values were considerably lower than when at 200 as a majority of the population could outlive the simulation. The grid size was also increased from the default value of 24 to 50. This change was made in order to minimize the volatility in average fitness between generations once the optimal chromosome was found which was essential for comparing changes to the genetic algorithm. This essentially lowered the probability of an unlucky generation. You can see this volatility drop in the graphs below.

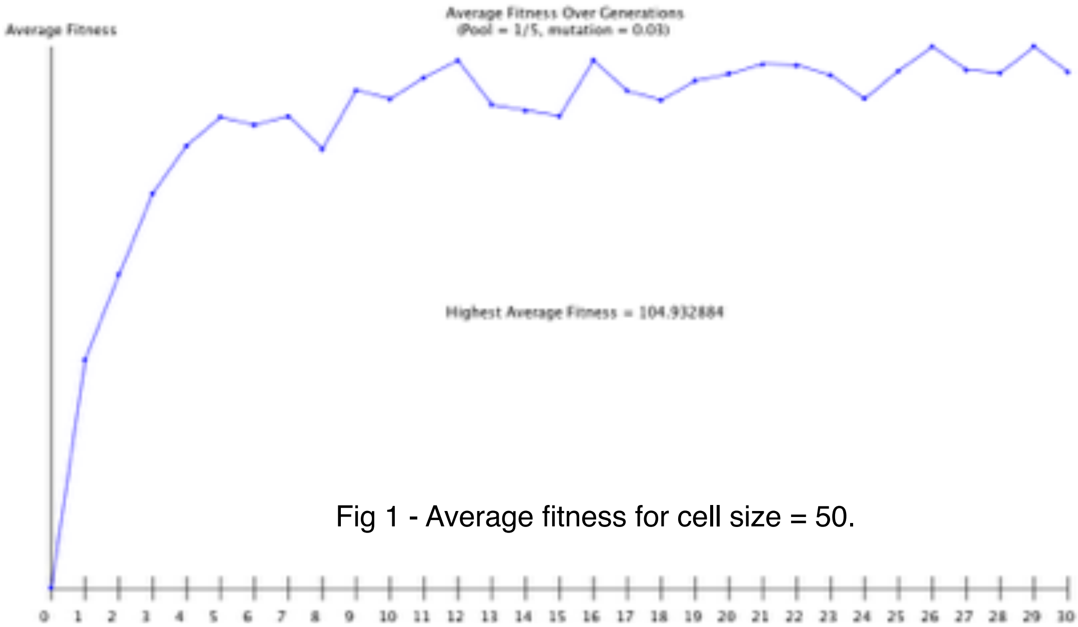

Fig 1 - Average fitness for cell size = 50.

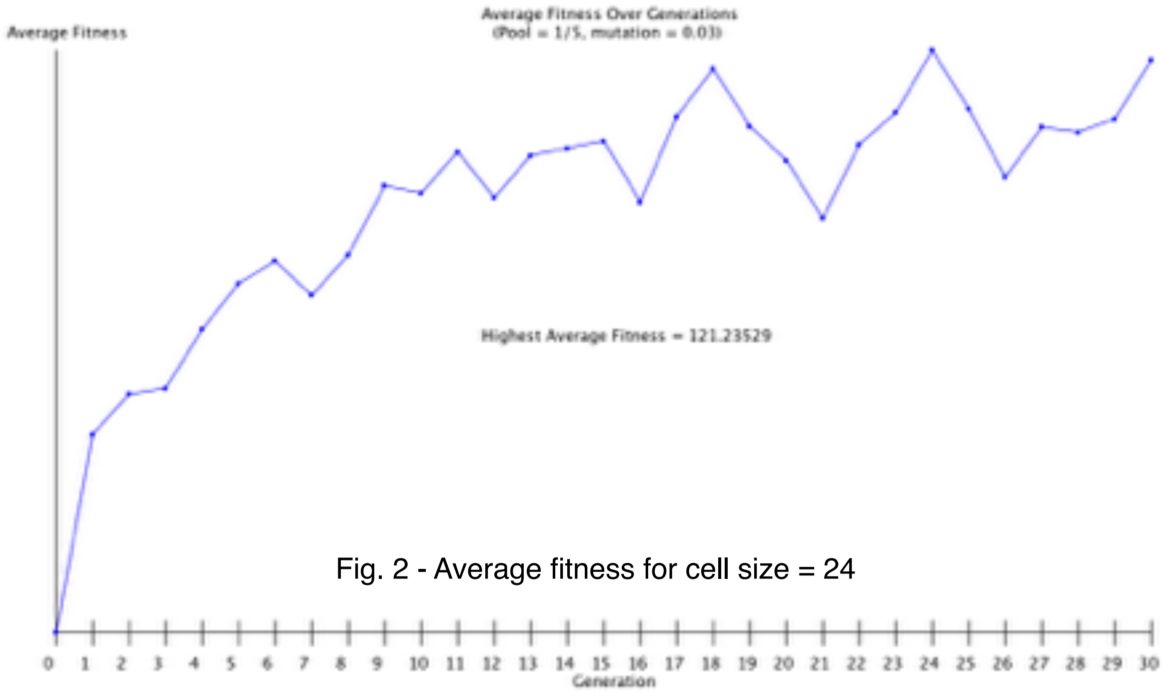

Fig. 2 - Average fitness for cell size = 24

Agent Function

For my percept format I chose to work with percept format 2. With the knowledge that the percept and action vectors were mapped identically and that opposite actions were mapped by (8-i) I set about a function that added individual weights to the corresponding seek and avoid actions. The function works by splitting the percepts into three sections of nine and performing different operations based on the location of the percept in these sections. The action vector would then contain all of the added weights from these sections allowing it to communicate a decision accordingly.

Section one pertains to monster movement and which would add the 'seek monster' and 'avoid monster' weights to the direction of the monster and direction opposite the monster respectively.

Section two pertains to creature movement and would add 'seek creature' and 'avoid creature' weights in the same way section one would for monsters.

Lastly, section three pertains to food actions. Any percept detected in this section would also have corresponding 'seek food' and 'avoid food' weights added. However, a special condition was introduced for when food was on the current square. This was done on the assumption that a creature would not try to eat otherwise. In this case if there was a value within percept 22 the eat action would have a weight added. The weight added for eating would also be multiplied by the value of the percept. This ensured that a creature would be greater inclined to eat when a strawberry was red (and had a percept value of 2) rather than when it was green (and had a percept value of 1).

Each time the agent function was called, a 'random action' weight was also added, but only once. This ensured that on the detection of low priority percepts or no percepts at all, the creature was likely to move randomly.

Chromosome

All the weights for the agent function were encoded in the chromosome. As such the chromosome was initialized as an array of floating-points, each with a value between 0 and 1. The small size of the chromosome made it easy to assess the priorities of any given creature. Because of this small size and its ease in understanding i could print the chromosome of the creature with the highest fitness and could predict its tendencies (though this is commented out in the final submission). This was ideal as it also made mutations in a given chromosome easy to find.

The Genetic Algorithm

For my genetic algorithm, the final version used tournament selection with a 1/5 population subset, a mutation rate of 3% and elitism by retaining the fittest individual each generation to ensure that progress by the algorithm was not lost. By experimenting with the mutation rate I found that changes made very negligible differences. I believe that this is due to the relative ease the model has in finding an optimal chromosome in the first place as more often than not a consistently successful chromosome would be found within the first 4 generations. For the act of

mutation, I decided that the mutation would change 2 values of the chromosome in order to increase the possibility of discovering different winning chromosomes. Randomness was also introduced within my crossover operations. Each crossover would begin at a random index, which is consistent with the idea that the algorithm would not know in what aspects one parent would outperform the other.
For parent selection, I chose to use tournament selection as I believed that the low probability of unfit individuals being selected in roulette wheel selection would result non-diverse populations which in turn would increase the chances of a population settling with a chromosome of a local, non-global maxima. To further ensure that non-diverse populations were not a factor in I also used a relatively low subset for tournament selection with 1/5 the size of the population. However when testing the algorithm at different subsets I found very few differences in my results as shown below.

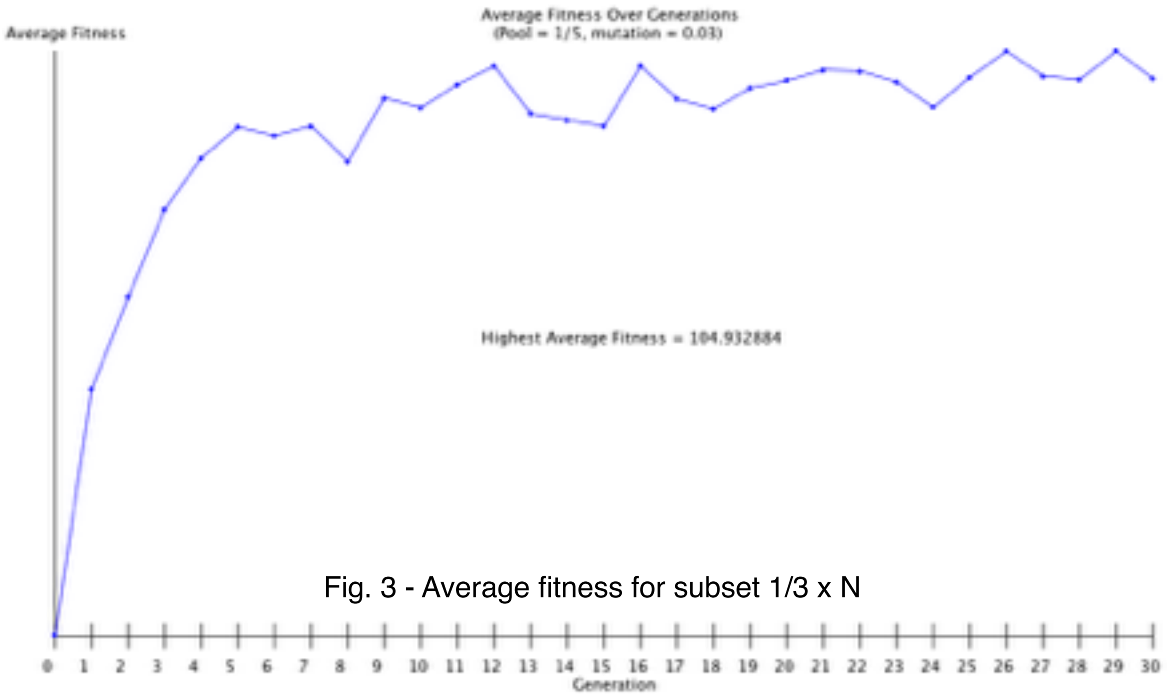

Fig. 3 - Average fitness for subset 1/3 x N

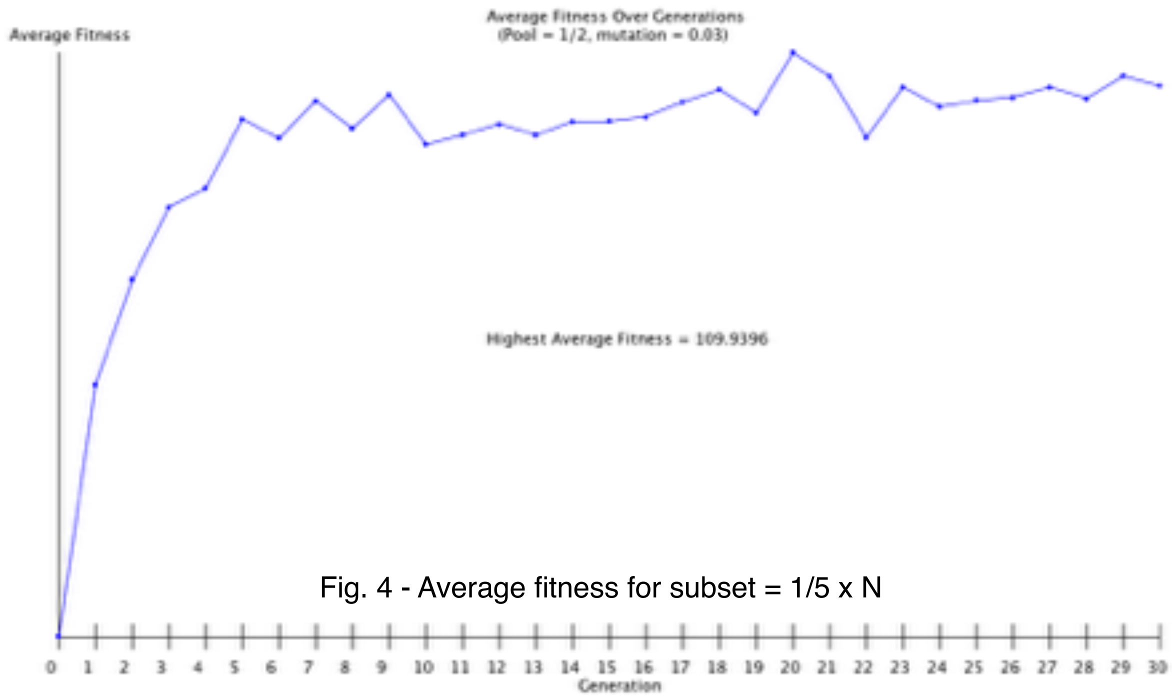

Fig. 4 - Average fitness for subset = 1/5 x N

For my fitness function, I began by using a simple addition of energy and duration of life but quickly found that this resulted in creatures with suicidal tendencies, that would seek monsters early in order to guarantee a high energy score. To balance this, I placed less of an emphasis on energy by dividing it's value by 5. This fitness function (duration of life + energy/5) gained much better results and hence was left as the final fitness function.

<u>Results</u>

The results of my genetic algorithm prove its success. When running on the default number of turns per generation, I was able to generate up to 70% survival. When the number of turns was doubled to account for the nature of my fitness function and the difficulty increased for the creatures i was still able to generate 20% survival rate within 30 generations. More often then not the winning chromosome had high low weights for monster seeking (< 0.1), high weights for monster avoidance (> 0.9), roughly even weighting for creature seeking and avoidance (0.5) and high and low weights for food seeking and avoidance, respectively.

The winning chromosome usually had a weight for eating that was just slightly lower than half it's monster avoidance so that when presented with a red strawberry on its square (2 x eating weight) and a monster in its vicinity, it would prefer to escape the monster than eat. On average, intelligent behavior would begin to be exhibited by generation 6 as seen on my graph of results below.