INFO 6205 Final Project – Generic Algorithm for *Robot Controller*

Group 215

Group Members:

PinHo Wang

PeiChun Tseng

Instructor: Robin Hillyard

**Implements**

***Problems:***

The problem is to design the robot controller to walk through the maze and arrive the destination in the end without crashing into the wall. The robot can be control by six different direction sensors, three on the front, one on the left, one on the right and one on the back. Each route can be one solution and can be scored, our purpose is to find the fittest solution by compare the scores until it reaches the termination we set.

We choose the highest fitness solution in each generation and change it by crossover and mutate for the next generation. The final solution will be found until the number of generations reach the maximum we set.

***Genotype***:

In individual class (Phenotype), we have one chromosome and its type is int[] with binary value 0 and 1. We call 0 and 1 value as gene.

***Expression***:

We define one chromosome as one solution. Each solution indicates a series of possible actions by binary code, such as 10011110…11. The definition of actions shows below,

00: do nothing, 01: move forward, 10: turn right, 11: turn left

***Phenotype***:

We dedicate Individual class as Phenotype, which contains one genotype.

***Fitness***:

The fitness of each solution is calculated by the route. We initially set the route we want and calculate the route provide by chromosomes. In each route, we calculate whether this route pass through the route we set, and it will be scored.

**Code**

*GeneticAlgorithm class*: The main GA implements.

*Individual class*: Details about individuals, including chromosome and fitness. It means one route solution in this problem.

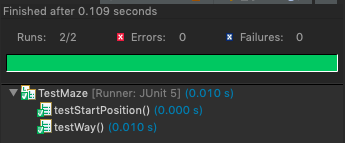
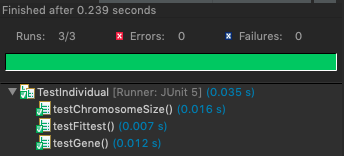
*Population class*: Details about populations, including individuals array.

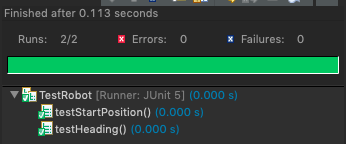
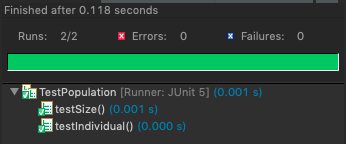
*Maze class*: Define the maze that contains the route and score the fitness.

*Robot class*: Details about all sensors and make the decision of the actions.

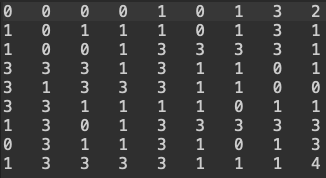
**Experiment**

***Test:***





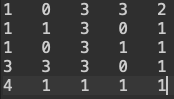
***Maze1:***

 Maze(9X9) Fittest Route

Best solution (30.0): 11111101110001001010101101111010101101101111101101000110010001101001010001100100110111100110010000110101110001010010000001011001

As the pictures show above, the left side is the maze, the right side is the route of the fittest solution after 1000 times generations. The result is pretty good that the robot follows the route we want exactly.

***Maze2:***

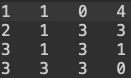
 Maze(5X5) Fittest Route

Best solution (8.0): 11111000010101100110011001101100111001110100111111110110000111101001000001010101101100101101011111011010110001111010111001001001

The result is quite corresponding the route we expect.

***Maze3:***

Maze(4X4) Fittest Route



Best solution (8.0): 10011111001010111011100111011011000001010010111110100111010001000111010001000110011001101011101001010110011101100000000100100100