Research

1.Genetic algorithms

Genetic algorithm is a heuristic (technique designed for solving a problem, or finding an approximate solution more quickly when classic methods are too slow or fail). Heuristic’s are in a way a shortcut to a solution yet using them forces us to lose accuracy or optimality we would have with other more classical methods. The use of this kind of method should be used in my opinion only when the problem is compatible , the solution we would get would still serve us even if it is not the most optimal, when the approximate timeframe in which the solution will be produced suits us. Genetic algorithm part of the Evolutionary algorithms that use techniques inspired by Biological evolution. GA’s specifically use techniques and manipulations that resemble natural selection in Nature. They start with an initial population of genes and play around with them using methods like “mutation” “fitness” “crossover” until an approximate solution is returned. Both in the start and during the journey to finding a possible solution a considerable amount of randomness is used and it should be considered beforehand whether randomness contributes or not to how fast a solution can be achieved or if the solution delivered will help us understand anything, better yet do we even need to understand anything about how the solution was found or just want a solution that works and that is it? Genetic algorithms are powerful when we know the approximate parameters of the problem, but we do not have enough information on how to pinpoint the solution without numerous iterations.

2. A\* search algorithm (<https://en.wikipedia.org/wiki/File:Astar_progress_animation.gif> )

A\* is an algorithm that excels in having the “best first search” to problems related to pathfinding and graph traversal. A\* is and informed search algorithm and that means it knows where its start point is, where the end point is, what are the boundaries of its environment and determine the path that will be the fastest to reaching its goal. To achieve this it iterates its main loop and determines which is the next best step to take based on the distance from the start (how much distance has been traveled) and the condition set by the heuristic *h(x) ≤ d(x, y) + h(y).* If the heuristic’s condition is met then A\* will find the path of least resistance to its goal. There are complexities with using A\* such as the priority queue that is used.

3. Conclusion

In the specific problem with a start and an endpoint on a grids that may or may not have obstacles the A\* will in almost every case outperform the GA due to several factors:

1. A\* is more closely specialized to pathfinding than GA.
2. A\* takes the path of least resistance to the target and does not take paths

that might lead it away from the goal.

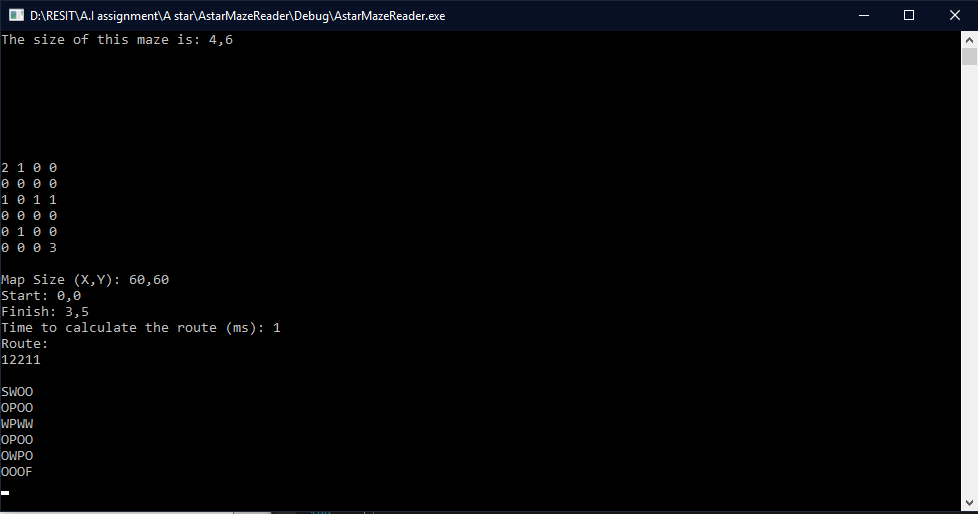
1. A\* does not create and destroy or manage as many objects or do as many calculations as GA.
2. GA might find a solution, but it wont the guaranteed best solution there is.

Code

There are 2 projects in my deliverables. One is a project that implements the use of A\* for solving the maze given and the other implements the use of GA to solve the same maze. Both are tested and the GA project has a user interface on the console. It prompts the user to give the parameters he wishes to use with the algorithm (chromosome count and number of moves given for solving the maze) and recommends parameters that are proven to work. To change what kind of maze is in used, the test maze must first be put where the project folder is and the code on line 175 (GA) or 252 (A\*) must be changed to match the name of the maze.

Tests

1. Testing A\*



Average time to calculate the route after several tries is: <= 1ms (instantly)

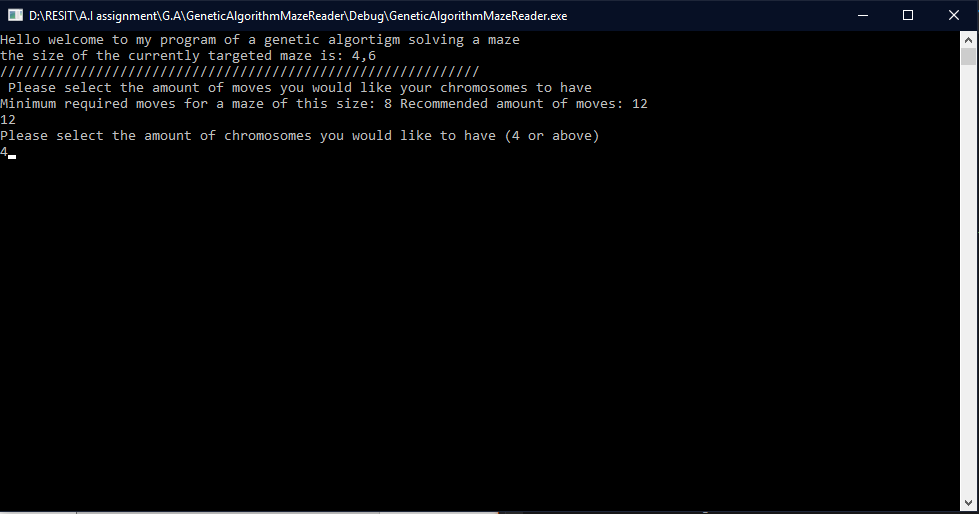
Route usually stays exactly the same.

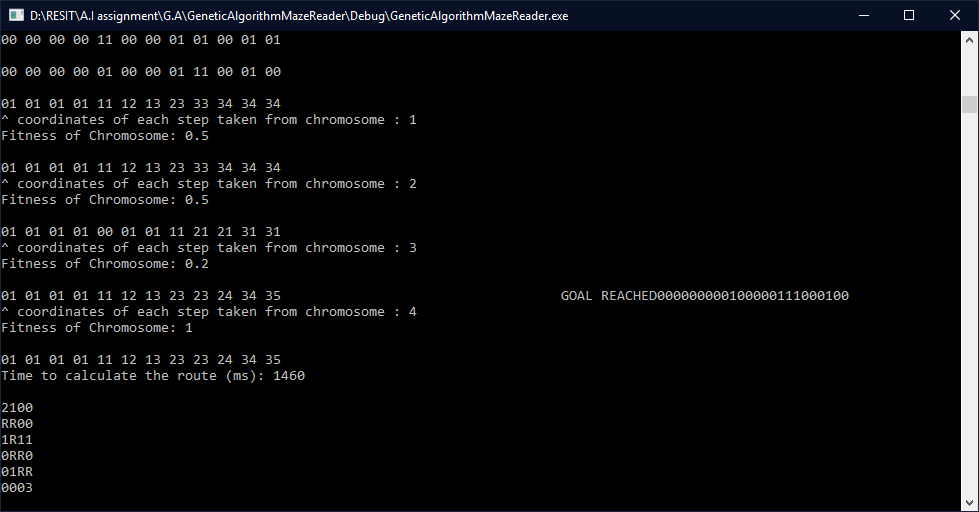
1. Testing GA

Test 1 (low chromosome count , but a lot of moves)

4 chromosomes.

12 moves allowed.



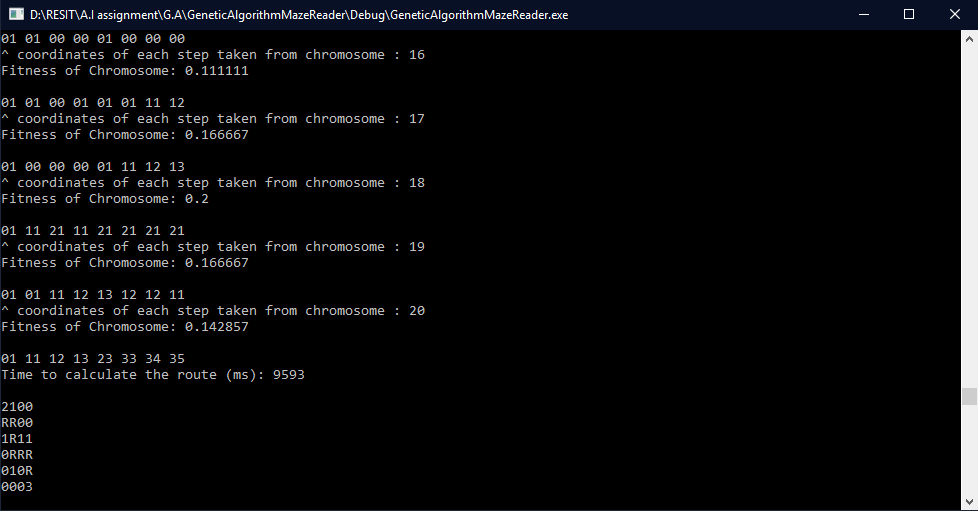


Time to calculate route : 1460 ms

Test 2 (high chromosome count, but less moves)

20 chromosomes.

8 moves allowed.

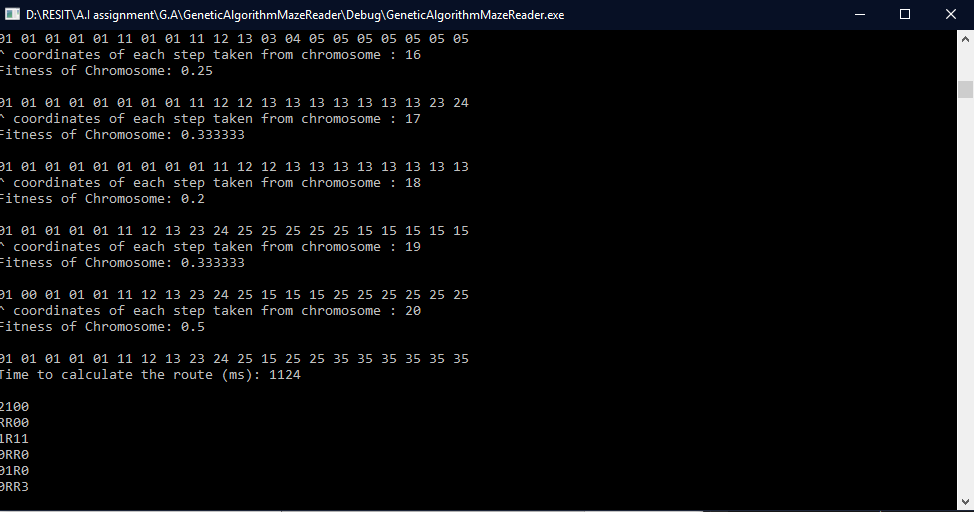


Time to calculate route : 9593 ms

Test 3 (High chromosome count and high move count)

Chromosomes 20.

Moves allowed 20.



Time to calculate route : 1124ms

Ideas for improvement

Improving GA’s fitness function to be better for the current problem

Testing out different priority queuing for A\*

Implementing a Generalised Adaptive A\* (GAA\*)