# Lab 3: Genetic Search Algorithms

1. Strategies
   1. Initialization – It creates a random population by randomly walking through the field and saving the steps taken, these steps are then used as the basis for selection for future generations attempting to optimize the number of steps taken to goal.
   2. Selection – This would be a variant of fitness proportionate and elitism, as the population is reduced to only the top half of performing chromosomes. Parents are then chosen randomly from this elite pool to fill out the rest of the population (the most elite from the previous generation are retained).
   3. Reproduction – Uniform Crossover method
   4. Mutation – Randomly select one step (bit) and flip it to a new random step
2. Minimum Possible Fitness Value – in this example, there is a -500 reward for reaching the end, a 500 penalty for a collision, and a penalty for each step taken. That means the path with the least cost is that which goes diagonally straight to the end without a collision, in which case the total fitness would be the number of steps taken (the dimension of the field – 1) – 500.
   1. 10x10 = -491
   2. 20x20 = -481
   3. 30x30 = -471
3. Ran with random inputs through consecutive iterations, full results can be found in log file on github repo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Size** | **#Generations** | **Population Size** | **Mutation Rate** | **Lowest Fitness** | **Lowest Fitness Reached** | **Method Timing** |
| 10x10 | 139 | 8 | .39 | -487 | Gen 42 | 563.65 ms |
| 20x20 | 82 | 27 | .28 | -413 | Gen 64 | 1335.3 ms |
| 30x30 | 82 | 27 | .28 | -271 | Gen 68 | 2328.5 ms |

1. See Code

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Size** | **#Generations** | **Population Size** | **Mutation Rate** | **Lowest Fitness** | **Lowest Fitness Reached** | **Method Timing** |
| 10x10 |  |  |  |  |  |  |
| 20x20 |  |  |  |  |  |  |
| 30x30 |  |  |  |  |  |  |