

# GENETIC ALGORITHMS TO SOLVE RADAR OPTIMIZATION PROBLEM

INFO6205-501

## **Team Member:**

Zigeng Fu. (001448458)

Yitong Liu. (001494304)

Feng Qiu. (001434172)



## Table of Contents

<b>1. Introduction .....</b>	<b>3</b>
1.1 Genetic algorithm.....	3
1.2 Project goal .....	3
<b>2. Question Description.....</b>	<b>4</b>
2.1 Case description.....	4
2.2 Euclidean distance .....	4
2.3 Terms .....	5
<b>3. Main Concepts .....</b>	<b>7</b>
3.1 Encoding .....	7
3.2 Decoding .....	7
<b>4. Parameter Settings .....</b>	<b>9</b>
4.1 Group size.....	9
4.2 Chromosome length .....	9
4.3 Mutation probability.....	9
4.4 Crossover probability .....	9
<b>5. Program Structure .....</b>	<b>10</b>
<b>6. Algorithm Design.....</b>	<b>10</b>
6.1 Data generator.....	10
6.2 Fitness calculation algorithm .....	10
6.3 Selection process .....	11
6.4 Crossover process .....	11
6.5 Mutation process .....	12
<b>7. Graphical user interfaces (GUI).....</b>	<b>14</b>
<b>8. Results &amp; Data Analytics.....</b>	<b>16</b>
8.1 Graphical Representation .....	16
8.2 Conclusion.....	16
<b>9. Testing .....</b>	<b>19</b>
9.1 Selection test .....	19
9.2 Crossover test .....	19
9.3 Mutation test .....	19
9.4 GA performance test .....	19

# 1. Introduction

## 1.1 Genetic algorithm

In computer science and operation research, a genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolution algorithm. The genetic algorithm is commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover, and selection. Our project is to use genetic algorithm to solve a complex military problem.

## 1.2 Project goal

In this project, our goal is to apply the Generic Algorithm on the position selection of a radar station. There are 15 military bases in an area, we need to find the optimal area where the radar station should be established so that the total signal strength is the best.

In the process of constructing the "RADAR OPTIMIZATION " model, we first built the offspring, genotypes, expressions, and the environment in the genetic algorithm. Then, the fitness calculation algorithm, the selection algorithm, the crossover algorithm, and the mutation algorithm are formulated. At the end we can get the optimal location of radar.

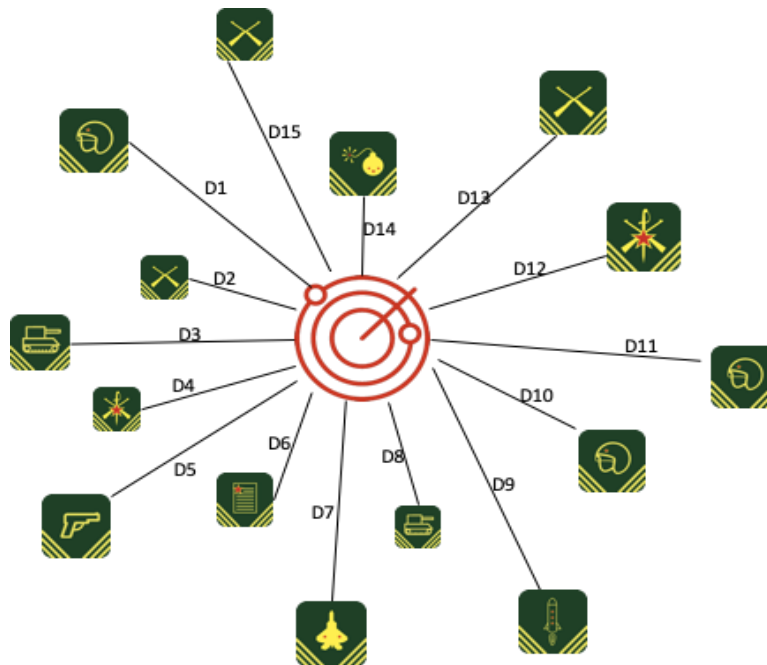
## 2. Question Description

### 2.1 Case description

In a country, there are fifteen military bases. Our goal is to establish a radar station so that the signals can meet the needs of all the 15 military bases shown as *Figure 1*.

However, considering the cost efficient and other factors, the strength of signals and the sum of distance from the radar to fifteen militaries is not a simple linear correlation.

Thus, we need to figure out the optimal location of radar according to the strength of signals by genetic algorithm.



*Figure 1*

### 2.2 Euclidean distance

First, we need to calculate the sum of Euclidean distance from radar to fifteen military bases, shown as *Equation 1*:

$$D = \sum_{i=0}^{15} \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}$$

Equation 1

Where D denote sum of distance,

$x_i$  and  $y_i$  denote the coordinate of one of the 15 known points,

$x_r$  and  $y_r$  denote the coordinate of the location of radar.

## 2.3 Terms

### 2.3.1 Individuals

An individual means a candidate location of the radar. The production of new individuals requires the reproduction of its father and mother, which are two other individuals. Each can inherit genes from parents.

### 2.3.2 Genotype

The format of the genotype will be presented as binary numbers with 20 bits (10+10). The genotype is divided into two parts, the higher 10 digits will be the X-axis, and the lower 10 digits will be the Y-axis. We can figure out the genotype of the 1024\*1024 points which the whole program based on. Genotype examples are shown as *Figure 2*.

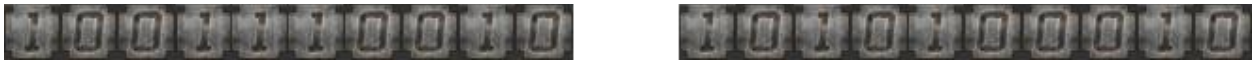


Figure 2

### 2.3.3 Expression

The expression will be presented in a three-dimensional coordinate system. The X-axis is the higher 4 digits of the genotype of the certain point, the Y-axis is the lower 4 digits of the genotype of the point, the Z-axis is the signal strength of the point.

### 2.3.4 Expected output

We are trying to find out the position which has the strongest signal regarding several factors such as the distance between the radar station and the fifteen pre-generated and fixed military bases.

#### 2.3.5 Fitness

In our model, we are looking for the point which has the maximum signal strength. The relationship between the stronger signal, the better fitness it has and the point has higher chosen probability. The point which has higher fitness should hold higher ratio to be chosen in the selection process.

### 3. Main Concepts

#### 3.1 Encoding

We assume a point (X, Y) as an independence. Then we use higher 4 digits represent value of X, and lower 4 digits represent Y so that we can represent every coordinate into an 8-digit integer. We store the encipher coordinates into an array, shown as *Figure 3*.

Encoding formula is:

Point (X, Y)  $\rightarrow$  integer =  $X*10000+Y$

Example:

P (155,350)  $\rightarrow$  int A =  $350*10000+155 = 3500155$



*Figure 3*

Data range of encoded value  $1024*1024=1$ million. So, the value of each coordinate is 1024, which will transfer it into a binary digit within 10 bits.

#### 3.2 Decoding

When we need to use a coordinate of a points, we have to decode the number into two parts, which higher 4 digits and lower 4 digits should be separated as following:

Integer  $\rightarrow$  Points (X, Y)

$X = \text{integer} \% 10000$  (Integer Modulo Operation)

$Y = \text{integer} / 10000$  (Integer Division Operation)

Example:

$A = 3500155 \rightarrow P(x, y)$

$x = 3500155 \% 10000 = 155$

$y = 3500155 / 10000 = 350$



## **4. Parameter Settings**

### **4.1 Group size**

1000 individuals

### **4.2 Chromosome length**

We have a total of 20 bits

### **4.3 Mutation probability**

0.02

### **4.4 Crossover probability**

The crossing must occur during reproduction, but the starting position of the intersection is entirely random.

### **4.5 Maximum of generation**

5000

## 5. Program Structure

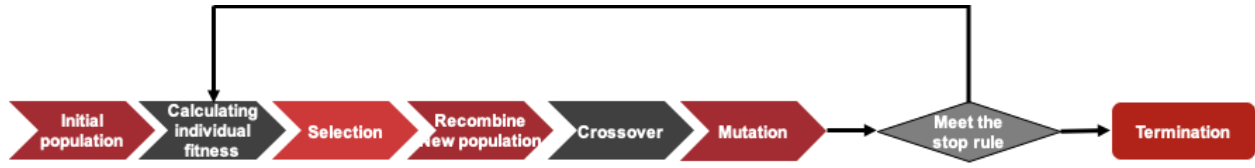


Figure 4

## 6. Algorithm Design

### 6.1 Data generator

Firstly, we pre-selected fifteen points randomly as our known fixed military bases from the 1024\*1024 grid.

Next, we randomly generate our initial group sample. Each integer contains the coordinate information of a point. The encoding and decoding method has been mentioned in Chapter 3.1.

### 6.2 Fitness calculation algorithm

For the 1000 individuals in each generation, we need to calculate the fitness of them for selection. In our case, the fitness represents for the signal strength at the specific military base. According to the signal propagation knowledge, we assume that the signal propagates as a sin waveform, and the signal wastage during the propagation is linear. So, we came out the original formula of fitness, which is a combination of a sin function and a linear function, shown as *Equation 2*.

$$S(D) = A * \sin(\omega D + \phi) - kD + b$$

Equation 2

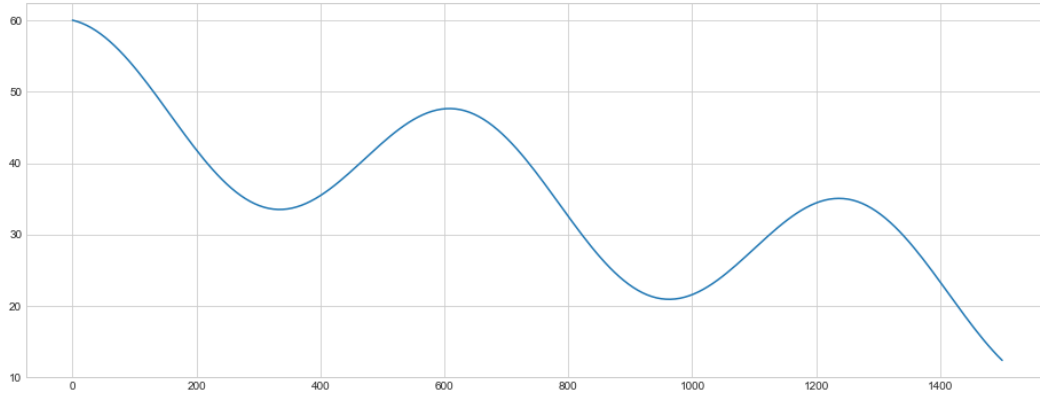
Then, we set the factors and figured out the final formula for fitness calculation given the Euclidean distance between the signal station and the specific military base, shown as *Equation 3*.

$$S(D) = 10 * \sin(0.01(D + 50\pi)) - 0.02D + 50$$

*Equation 3*

Where D for Euclidean distance and S(D) for the signal strength, i.e. fitness.

The functional image of *Equation 3* is shown as *Figure 5*.



*Figure 5*

### 6.3 Selection process

We use Roulette method to achieve the selection process. The basic idea is that the probability that each individual is selected is proportional to its weight.

We firstly assign weights for 1000 individuals in a generation according to their calculated fitness. The weight here represents for the selection probability. The higher the fitness, the higher the weight the individual takes up. And the sum of weight is 1.

Thus, in each evolution, based on the current generation, we can select the 1000 individuals of the next generation using Roulette described above.

### 6.4 Crossover process

The crossover is a process of individuals' reproduction. Traverse the whole group and do the following actions for each individual. Before doing the crossover process, we need shuffle the 1000 solutions so that the two solutions  $i$  and  $i+1$  we pick out later to crossover will be randomly selected.

Crossover steps:

- Randomly choose two Individuals to A and B to crossover
- Randomly generate the target-mutated bit M from 0 to 9
- Randomly choose to exchange the coordinate in X-axis or Y-axis
- Exchange the Mth bit of A and B, which is shown above

Crossover Example:

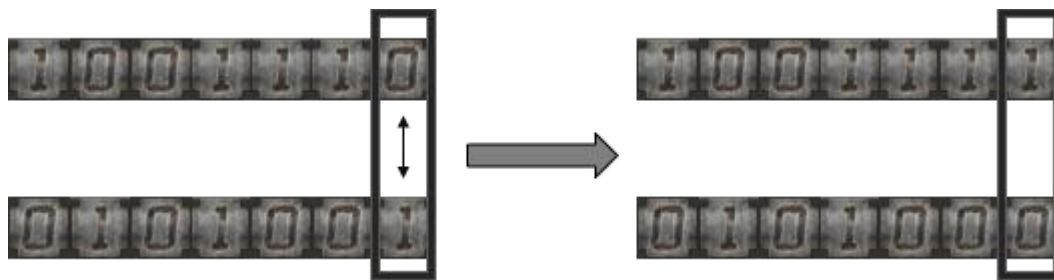


Figure 6

## 6.5 Mutation process

Mutation is a genetic level variation, not for individuals' level. Throughout the whole group, each's genetic sequence is likely to mutate over time. However, this mutation is not the entire gene sequence replaced by a new gene sequence, but some sites on the gene sequence been randomly replaced, and the replacement of these sites does not necessarily lead to changes in phenotype.

First, select the position to mutate from 20 bits recorded as tmpLoc. This process also randomly decides either X-axis or Y-axis to exchange.

Then let the gene mutation rate be 2% by selecting 1 from 50 integer. When mutation happening, if the value on tmpLoc equals 0, then change this bit into 1, else if the value equals 1 change it into 0.

Mutation Steps:

- Randomly generate an integer from 0-19 as the target-mutated bit M

- If  $0 < M < 9$ , we change the Mth bit of x coordinate
- If  $10 < M < 19$ , we change the  $(M-10)$ th bit of y coordinate
- Transfer from 1 to 0, or from 0 to 1

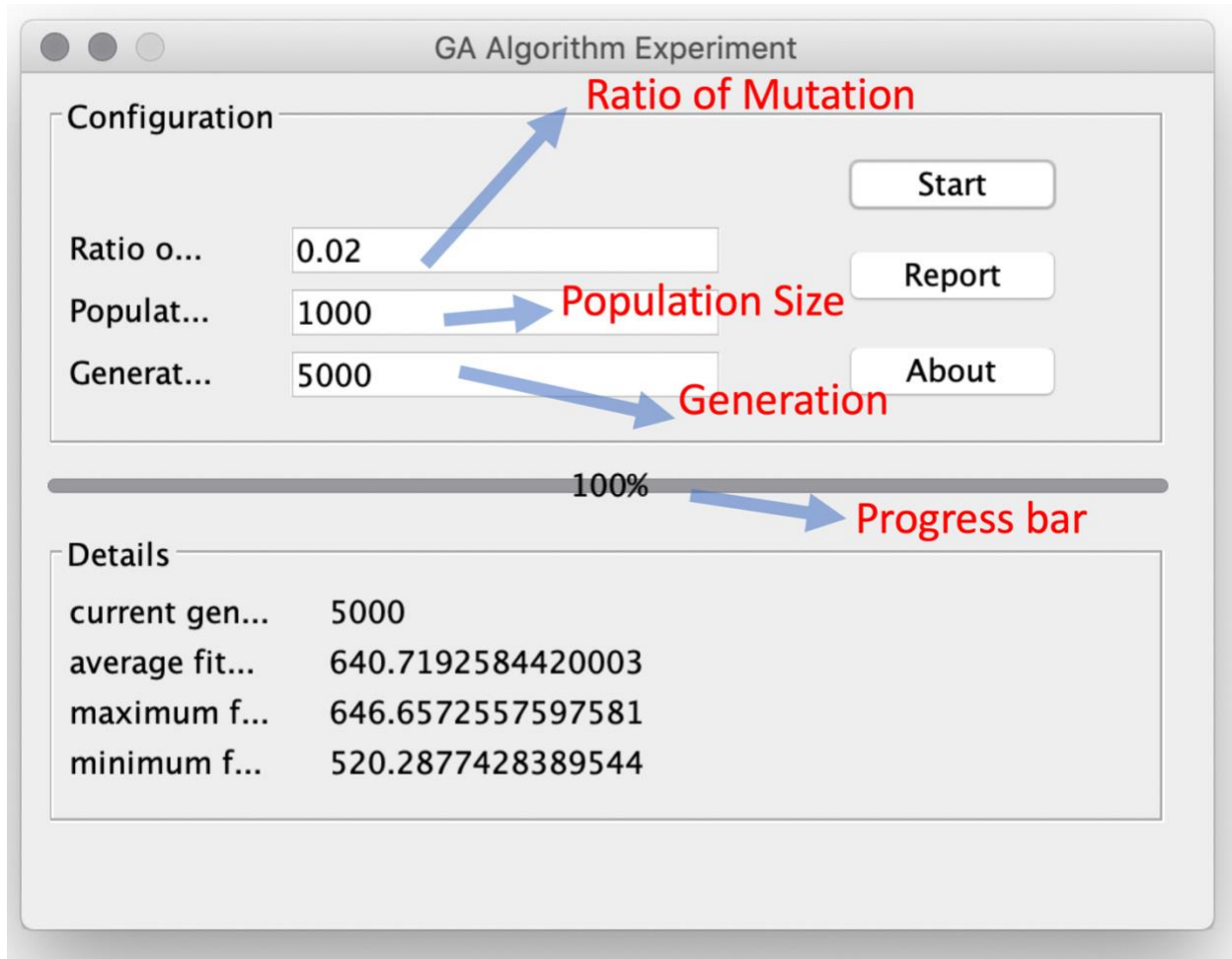
Mutation Example:



*Figure 7*

## 7. Graphical user interfaces (GUI)

This is the GUI of our program. The pre-defined parameters are shown in the text fields, and user can observe the progress of the evolution, shown as *Figure 8*.



*Figure 8*

At the end we are able to generate a report. This report shows the distribution of fitness for each generation. As the increases of generations, the degree of fitness gradually increases. When reaching a certain generation, the fitness tends to be stable, shown as *Figure 9*.

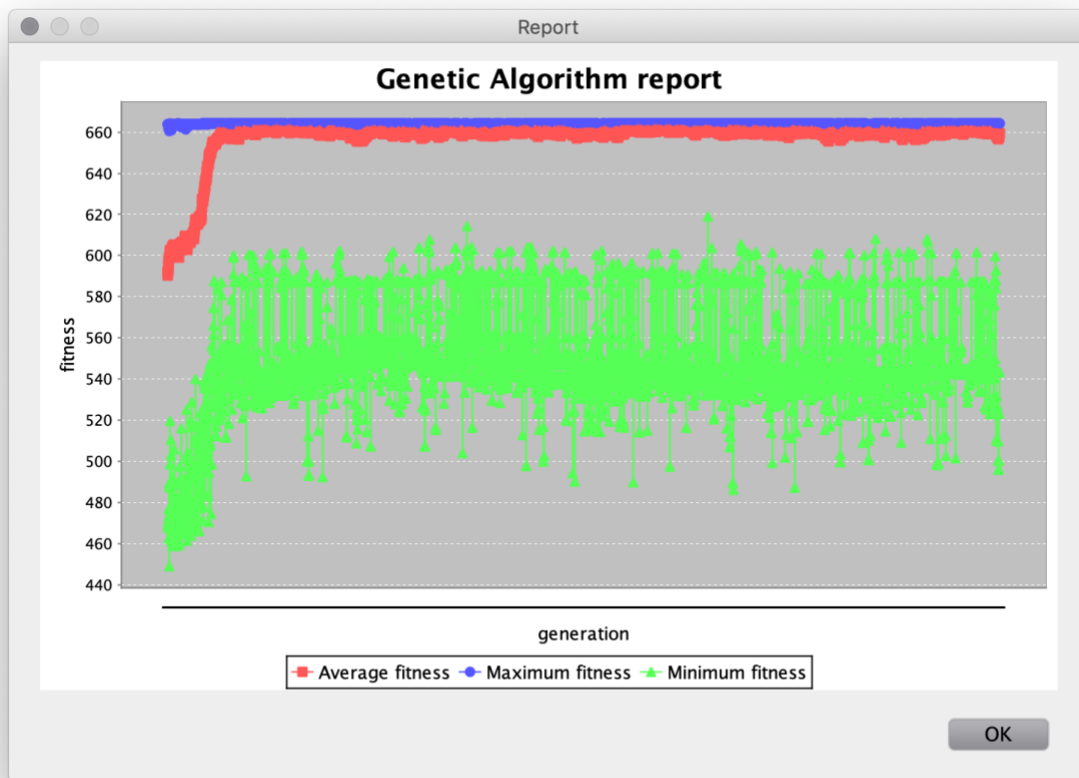


Figure 9

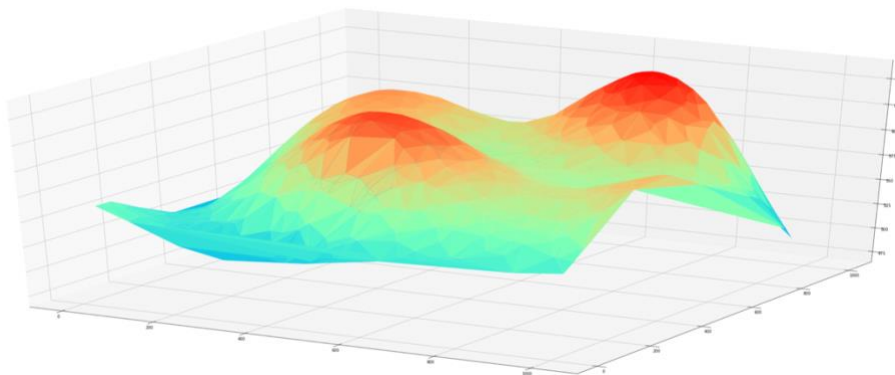
## 8. Results & Data Analytics

### 8.1 Graphical Representation

As we can see from the graph, the solution has several peaks. The bigger Z-axis value, the stronger signal strength, the brighter as shown on the graph, the better to select as our optimal points, shown as *Figure 10*.

- X-axis: the higher four digits of genotype before converting into binary numbers.
- Y-axis: the lower four digits of genotype before converting into binary numbers.
- Z-axis: the strength of signal.

Colored points: 1000 ethic groups evolved according to genetic algorithms.



*Figure 10*

### 8.2 Conclusion

After 5000 times selection, crossover and mutation, the 1000 final results, which are red points now, convergence in one limited range.

The graph demonstrate that we have three higher peaks in the whole solutions group. Which means, the ideal point we are looking for should land up on one of these lower peaks, shown as *Figure 11*.



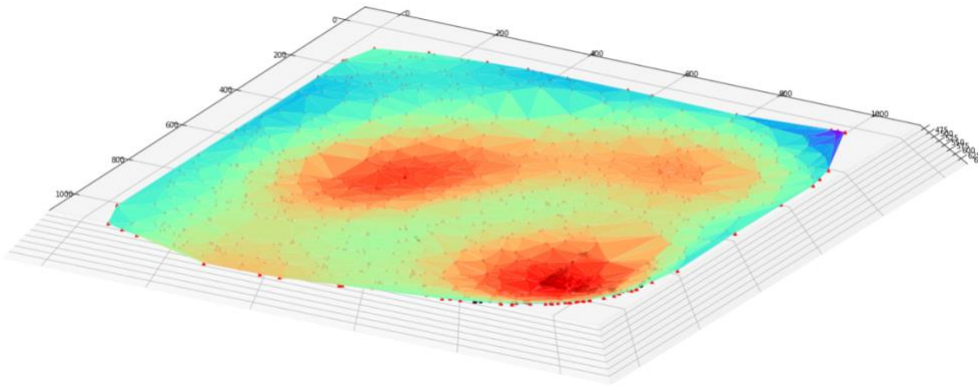


Figure 11

As we can see from the graphs, most of the red points (which is solutions calculated by genetic algorithms) are gathered in the highest peak. Which means, our genetic algorithms generally perform well. And the solution is around (950, 975), shown as Figure 12 &13.

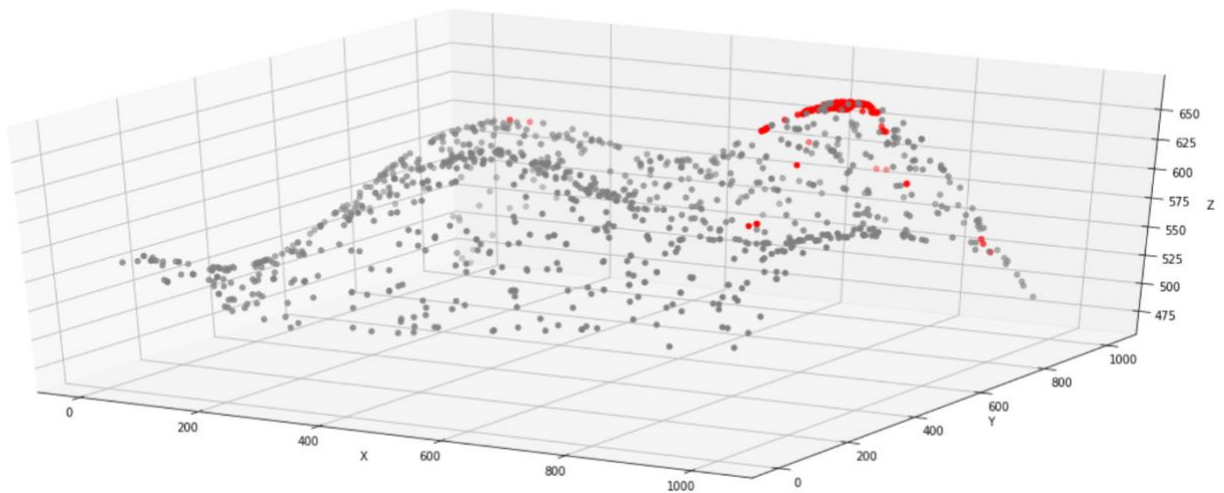


Figure 12

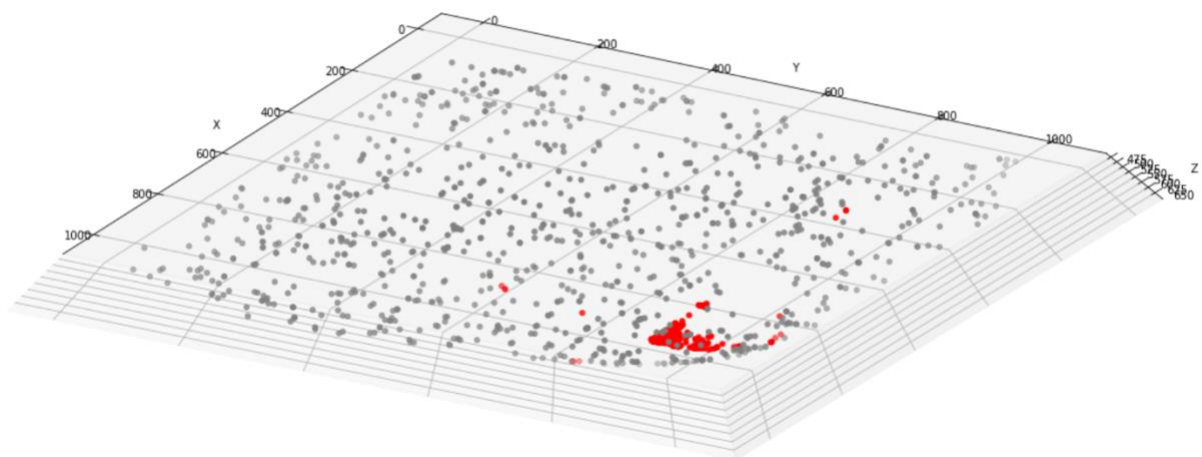


Figure 13

## 9. Testing

We finished the unit tests including the selection function test, crossover function test, mutation function test, and the overall GA performances test.

### 9.1 Selection test

- Initialize the first generation and calculate the fitness of each individual.
- After using the selection method, check the probability of each individual.
- If the fitness value is bigger, the possibility it will be chosen is bigger.

### 9.2 Crossover test

- The first case: binary code of a is 00011111 and binary code of b is 10001111. Crossover the first bit of these two codes and the crossover result should be 10011111 and 00001111. Then the decimal code of 10011111 is 159 and Decimal code of 00001111 is 15.
- The second case: binary code of c is 00001001 and binary code of d is 01110111. Crossover the third bit of these two codes and the crossover result should be 01010111 and 00101001. Then the decimal code of 01010111 is 87 and Decimal code of 00101001 is 41.

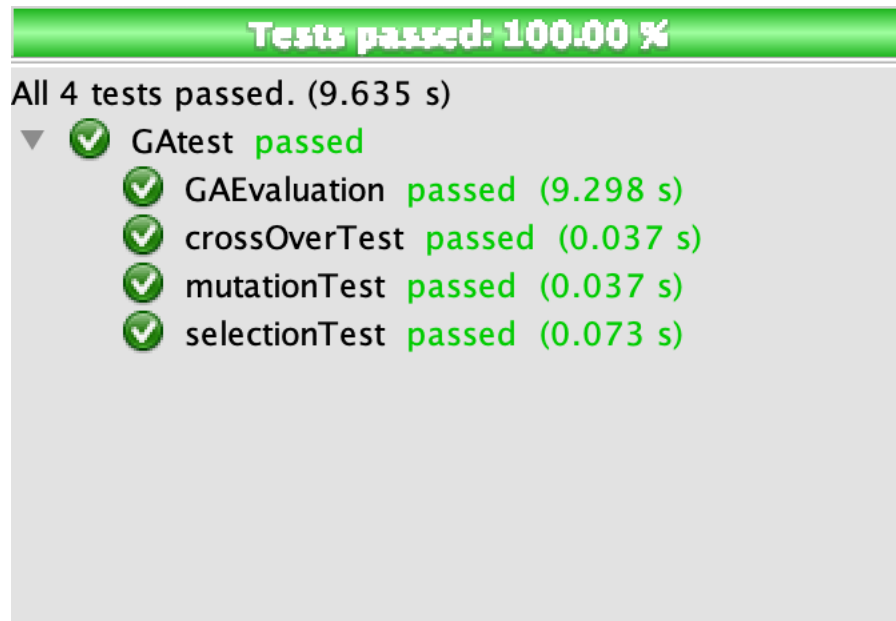
### 9.3 Mutation test

- The first case: the binary code of a is 00101001 and mutate the first bit of it. The result is 10101001 and the decimal code of it is 169.

### 9.4 GA performance test

- Initialize the first generation and calculate the sum1 of their fitness value.
- After 5000 times of selection, crossover and mutation, calculate the sum2 of their fitness value.
- Compare these two values, if sum1 is smaller than sum2, GA has a positive performance.

The test results are shown as *Figure 14*.



*Figure 14*