Game of Life

Genetic Algorithm

INFO6205_116

Team Member:

Yue Liu - 001353606

Maqi Shi - 001057366

1. Introduction

1.1 Genetic Algorithm

In computer science and operations research, a genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. In this project, what we need is to understand how genetic algorithm and use it in problem solving.

1.2 Problem Realization

In the process of constructing the Game of Life, we built the offspring, genotypes and the environment in genetic algorithm. Also, we designed the algorithm of fitness, the crossover algorithm, the evolution algorithm and the mutation algorithm for our own genetic algorithm.

2. Question Description

The Game of Life is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves. For this project, we need to use genetic algorithm to find an infinite growing and acyclic pattern for this game.

3. Parameter Setting

3.1 Group size

40*40 for 160 individuals

3.2 Chromosome Length

One chromosome has 2 genes

3.3 Gene

The gene length is 10
The first gene is random number between -20 and 20

3.4 Mutation Probability

Random number between 0.05 and 0.1

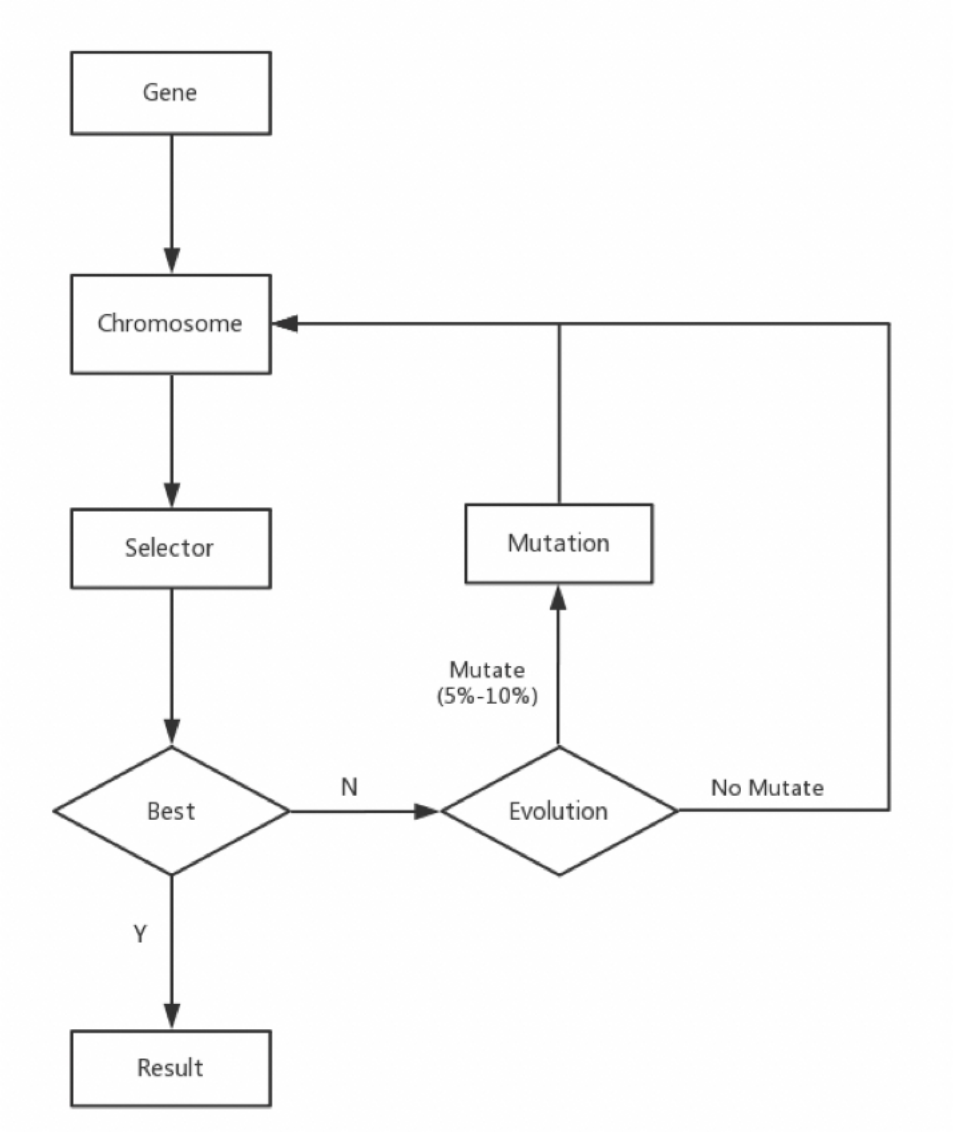
3.5 Probability of Death

Only the best can live, others will die

3.6 Maximum number of generations

The max number of generations is 1000

4. Program Structure



5. Algorithm Design

5.1 Gene(Genotype) Creating

We created a group of genes by giving them random integers. Each gene is an array contains 10 integer numbers.

5.2 Chromosome Building

Based on the definition, we use genes to build chromosomes (In this project, each generation of chromosome's population is 500.). One chromosome contains two genes, x gene and y gene, (x, y) make a coordinate represent for the live cell. One chromosome is one pattern, and the chromosome can compile genes and return a string of coordinate.

5.3 Behavior(Phenotype)

The behavior contains two important properties. The first is "generation", it represents the generation a chromosome can get under the rule of Game of Life. The second is "reason", which represents the reason of the death of a chromosome under the rule of Game of Life. "0" means that the cell will go extinct, "1" means a repeating sequence will be noted and "2" is the maximum configured number of generations will be reached.

5.4 Best Chromosome Selection

The next step is to select the best chromosome. By using a loop, the iterate chromosome get its own behavior and compare the behavior to each other to get the best one. The criteria of the selection are fitness. The chromosomes' fitness are generation numbers. The standard of the selection is to find the chromosome with the best fitness.

5.5 Fitness

Under our definition, fitness is the generation a pattern (a chromosome) can get under the rule of Game of Life. In this project, fitness is the property "generation" comes from object "behavior" of chromosome.

5.6 Evolution Algorithm

For evolution, the best chromosome's genes are used to create child genes. In the evolution function, a gene will change. The rule is that, if its value > 0, it will be decreased a random number between 0 and 10, and if its value < 0, it will be add a random number between 0 and 10. If it's value = 0, it remains the same. The purpose of our design is to make more interaction for genes so that they can get chances to reproduce.

5.7 Mutation Algorithm

During the evolution, it has 5%-10% possibility to have a mutation, which means it will be add a random number between -20 and 20 in the gene. Then, it will keep going

what we described before.

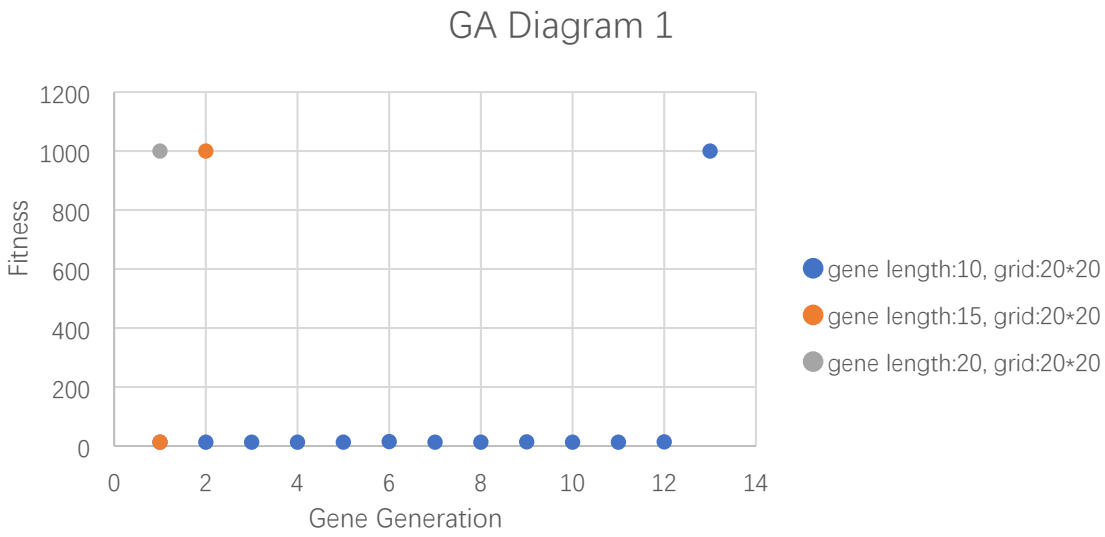
5.8 Result

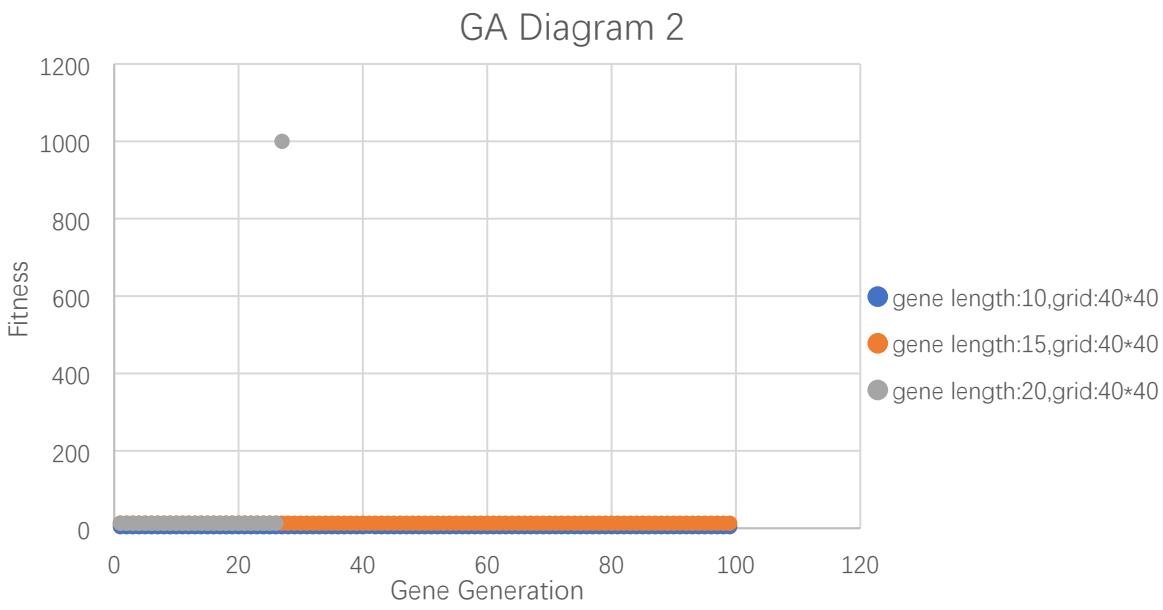
If we find one best chromosome which can have 1000 generation in behavior, that means it is the pattern that we are looking for. And if the algorithm cannot give out a result after 500 loops' running, it will return the best chromosome we can get in this term's running.

6. Data Analysis

We have test 6 sets of data, which are:
1) 20*20 grid, 10 numbers in one gene
2) 20*20 grid, 15 numbers in one gene
3) 20*20 grid, 20 numbers in one gene
4) 40*40 grid, 10 numbers in one gene
5) 40*40 grid, 15 numbers in one gene
6) 40*40 grid, 20 numbers in one gene

We set them into 2 groups based on the grid size and make two diagrams so that we can see the changes directly.





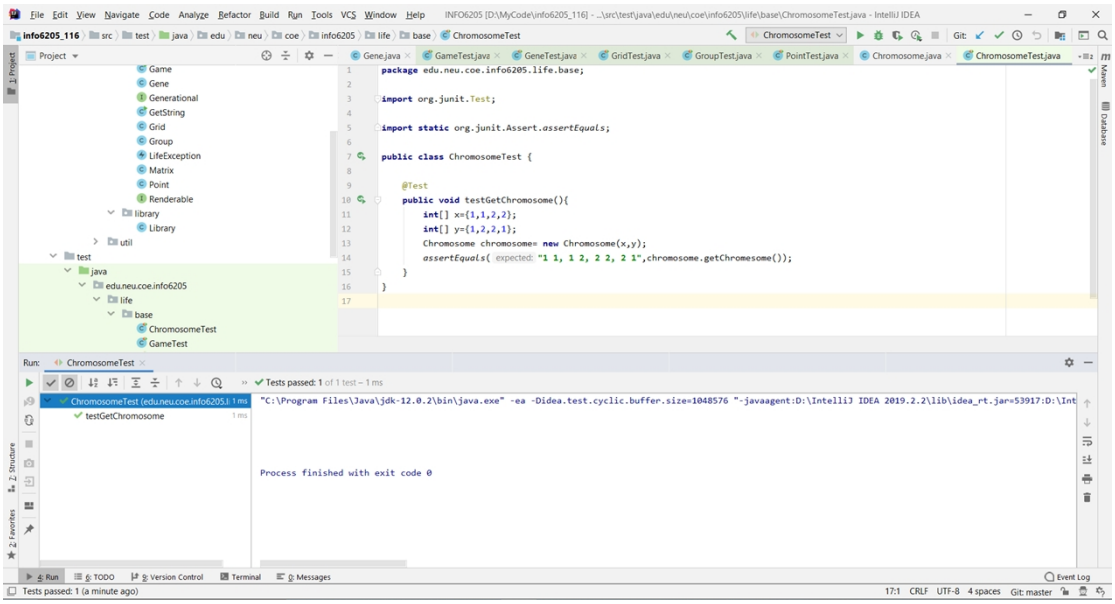
7. Conclusion

From the diagram we can see that, to find the best evolution pattern, the key data is the density of the genes. From Diagram 1, it is clearly that with the growth of gene length, the chance to find the best pattern is growing. And for Diagram 2, we can see in the grid of 40*40, only under the length of genes is 20 that can we find the best patterns. In conclusion, the density of the population plays an important role in evolution. The high density means that the genes will get more chances to “meat” each other, and that make the genes get more chances for interactions to give out more reproductions (based on the rule of Game of Life).

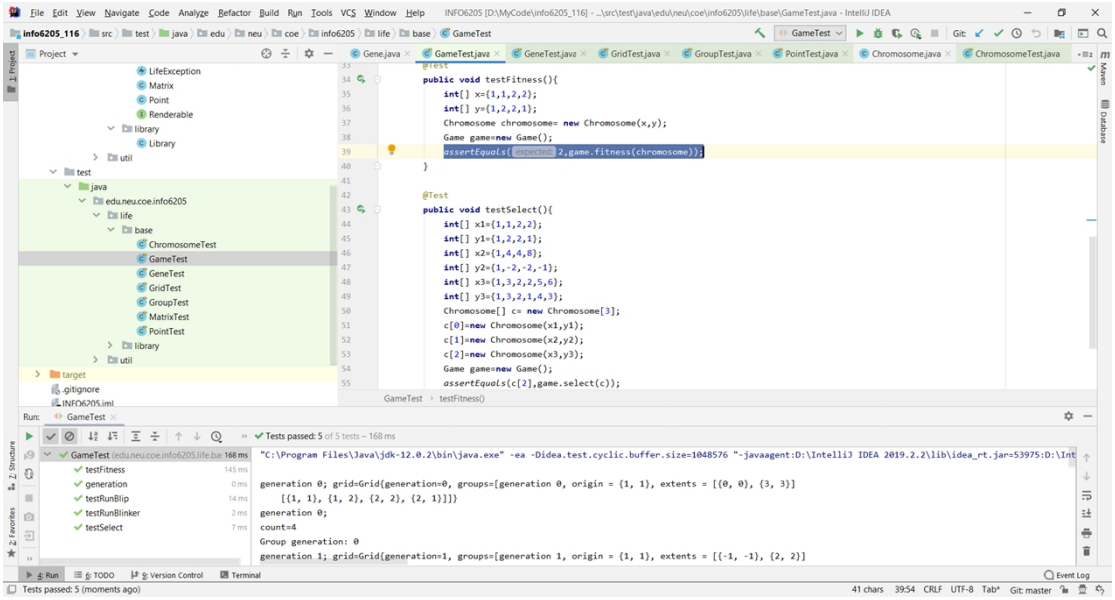
And the interesting thing is that, for those results that cannot get an infinite growing and acyclic pattern, the commonness is that, they will either die or get the cyclic patterns.

8. Test

8.1 Chromosome Test



8.2 Fitness Test



8.3 Evolution Test