December 7, 2019

Game of Life

INFO6205_112

Team Member
Han He 001824967
Yidan Sun 001059769

Contents

1. Introduction

1.1 Genetic Algorithm

In computer science and operations research, a genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). 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. Our project is mainly about finding the best solution to increase the number of evaluate generations with the gene pool.

1.2 Project Goal

In this project, we need to generate our starting patterns via genetic algorithms randomly. Applying Genetic Algorithm to find the best genotype. We should select the best genotype from the original population at first. Then applying mutation operation and the same fitness to find the best mutation genotype.

During the process, we need to set the number of evaluate generation sand maximum number of generations first. Then, we should calculate the fitness of each individual in initial graph groups. After selection, crossover, mutation and increase diversity operations, the best generation of groups is obtained. We also need to set termination conditions for judgment. Before that, we will output the graph with maximum fitness obtained during the evolution as the optimal solution.

2. Problem Statements

2.1 Overview

All biological species evolved from a small number of common ancestors after a long natural selection process. We hope that through the method of selecting the best, the children of the common ancestor can reproduce for a longer time in the same living environment.

2.2 Related Concepts

2.2.1 Individuals

An individual means a candidate with 2-Dimensional graphic and its location. Each shape is depicted on a 2-dimensional rectangular coordinate system, determined by its chromosome. The number of initial individuals is 10.

2.2.2 Genotype

The format of the genotype will be presented as a string with 50 Integers. The genotype Integers would be in the domain of [-20, 20]. Genotype will be used to mutate and inherit to the next generation.

2.2.3 Phenotype

The 2-Dimensional graphic composed of 25 point pairs is the phenotype. The extent of these points would be (-20,-20) to (20,20). The position of the point is rendered as a graphic, and the graphic will not be inherited.

2.2.4 Expression

The expression will be presented in a 2-Dimensional coordinate system.To be specific, two integers in genotype determine the coordinate of one point in phenotype, which used to to construct the current generation's shape. The console displays the genotype of all current individuals. Includes the genotype of the best one of initial 10 individuals and the individuals with the best fitness after countless mutations. The UI shows the phenotype of the individual which fitness is better than before.

2.2.5 Fitness

The fitness argument in our system is how many generations that an individual graphic can evolute. The larger fitness means the corresponding initial graphic fits the environment better, or in other words, more competitive.

3. Parameter Settings

3.1 Group Size

We generate the group size with 10 individuals. Each individuals has 25 points. During the first mutation, our start pattern with the best fitness will generate 10 new individuals. We select the best pattern to do the second mutation operation. Each individual will have 100 offsprings and repeated this operation till we find the best pattern.

3.2 Chromosome Length

We generate Chromosome length as 25 point pairs at first. Their children may be change the Chromosome length during the second mutation.

3.3 Mutation Probability

First time mutation:

The probability of Mutator is 0.2

The probability of Single Point Crossover is 0.25

Second time mutation:

There is a certain probability that points will mutate and individuals will increase or lose points.

3.4 Maximum of Generations

1000

4. Program Structure

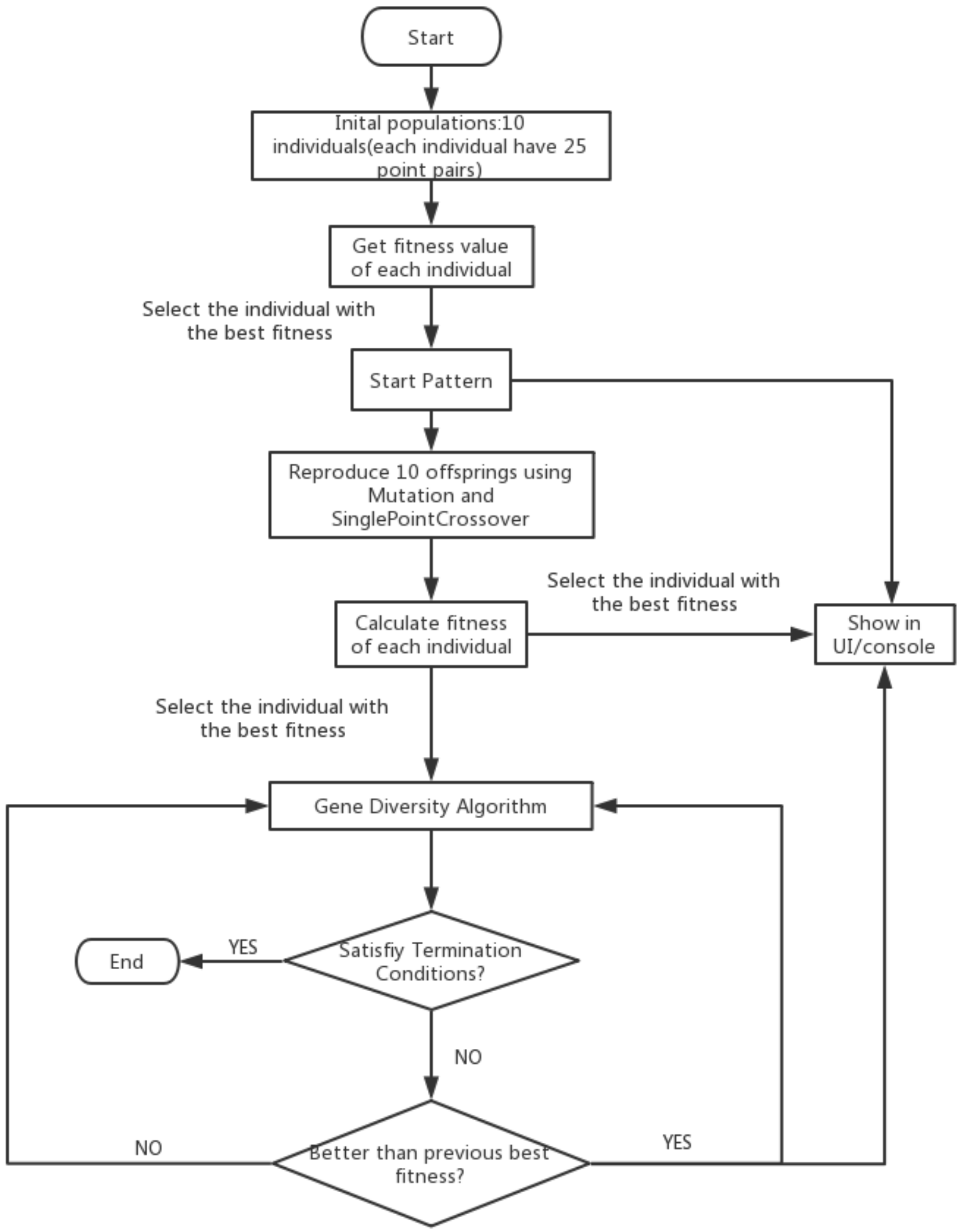

Figure 1

5. Algorithm Design

5.1 Calculation Algorithm of Fitness

The fitness function simply defined is a function which takes a candidate solution to the problem as input and produces as output how "fit" our how "good" the solution is with respect to the problem in consideration. We use the same fitness to find the different generation of the genotypes.

5.2 Terminate Algorithm

There are three conditions to terminate individuals' evaluation.

1. Individuals whose generation is over 1000.

2. All points in current individual is extinction.

3. If the shape of current individual is the same as fathers', it would terminate.

5.3 Start Pattern Algorithm

1. Stochastic process: 25 points are randomly generated in the range [-20, 20], and this process is repeated 10 times.

2. Set the control group: calculate the fitness of each individual in the initially generated graph group without any mutation operations. Find the individual with the greatest fitness and export the individual's chromosomes.

3. Generate the start pattern: the individual with the greatest fitness will serve as our initial pattern.

5.4 Single Point Crossover Algorithm

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. Crossover is sexual reproduction. Two strings are picked from the mating pool at random to crossover in order to produce superior offspring. The method chosen depends on the Encoding Method.

Single Point Crossover : A crossover point on the parent graphics is selected. All data beyond that point in the graphics is swapped between the two parent graphics. Strings are characterized by Positional Bias.

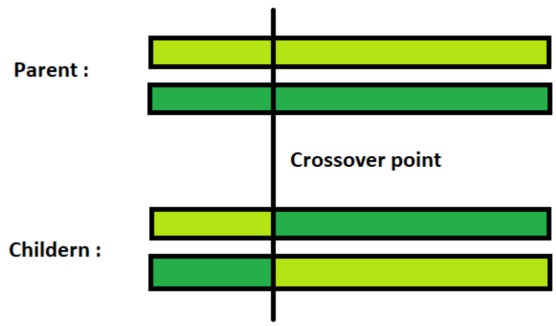

Figure 2

5.5 Mutation Algorithm

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.

5.6 Increase Gene Diversity Algorithm

Gene mutations are universal, random and beneficial in nature. The consequence is that a new gene appears instead of the original gene so the performance of the offspring suddenly appears a new trait that the ancestor never had. So in our project, our individuals also undergo mutations to increase gene diversity. There is a certain probability that the point will be mutated. Individual points will increase or disappear.

5.7 Terminate Iteration Conditions

When we select the individual with the best fitness from first time mutation. The second mutation will begin endless until it satisfy the terminate conditions. The following are the conditions:

1. When we have found an individual with a fitness of 1000 the iteration will stop because in terminate algorithm individuals whose generation is over 1000 will terminate evaluation.

2. All individuals are extinct.

6. Data Analysis and Results

The results are showing as below:

1. In most cases, we can get our best pattern with fitness greater than 1000 through our mutation operations.

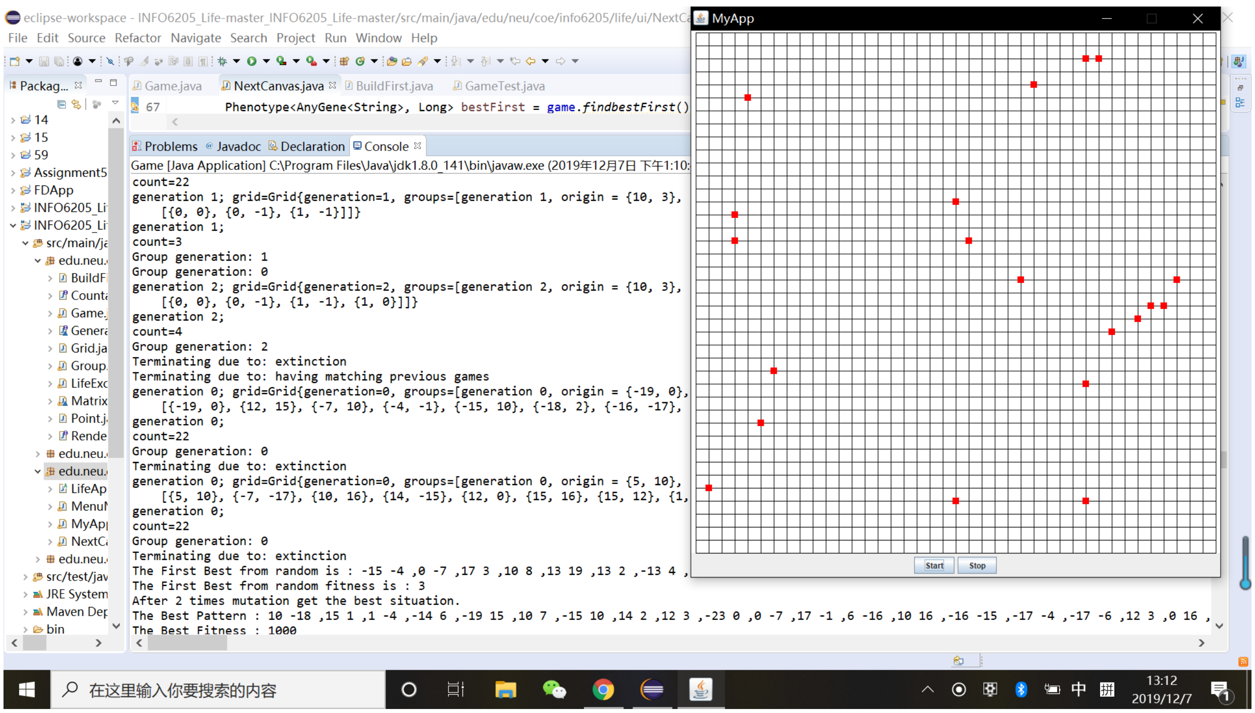

Figure 3

The First Best from random is : -15 -4 ,0 -7 ,17 3 ,10 8 ,13 19 ,13 2 ,-13 4 ,5 -15 ,18 10 ,12 -20 ,-14 -6 ,-14 -14 ,9 18 ,1 -18 ,18 10 ,6 -1 ,16 11 ,-13 2 ,9 2 ,-14 14 ,-19 16 ,12 -4 ,6 1 ,0 -15

The First Best from random fitness is : 3

After 2 times mutation get the best situation.

The Best Pattern : 10 -18 ,15 1 ,1 -4 ,-14 6 ,-19 15 ,10 7 ,-15 10 ,14 2 ,12 3 ,-23 0 ,0 -7 ,17 -1 ,6 -16 ,10 16 ,-16 -15 ,-17 -4 ,-17 -6 ,12 3 ,0 16 ,5 -1 ,16 1 ,11 -18

The Best Fitness : 1000

2. Another situation is that all individuals extinction after iteration. Then, the iteration will stop. So we will print out the best pattern who had the best fitness before.

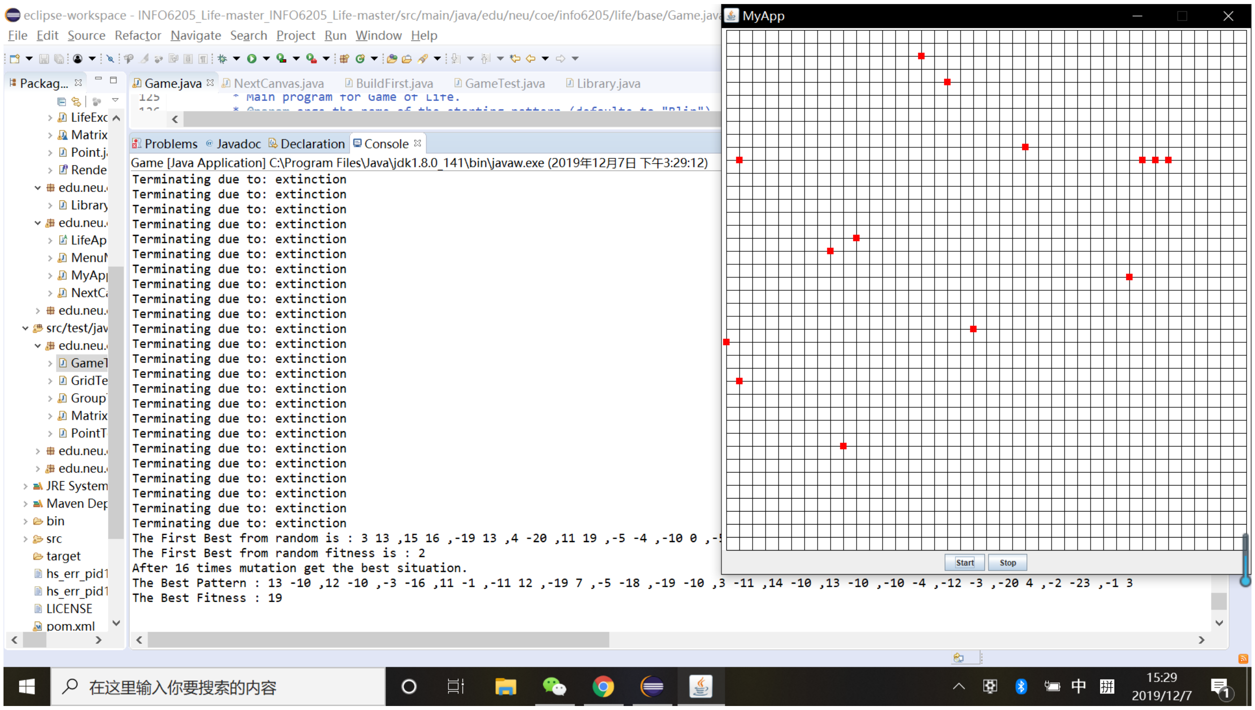

Figure 4

Terminating due to: extinction

The First Best from random is : 3 13 ,15 16 ,-19 13 ,4 -20 ,11 19 ,-5 -4 ,-10 0 ,-5 -3 ,14 2 ,-8 14 ,-1 -16 ,3 -14 ,7 1 ,-7 -4 ,3 2 ,-8 -9 ,-7 -13 ,-18 -15 ,17 12 ,4 -4 ,12 -15 ,-8 1 ,14 -1

The First Best from random fitness is : 2

After 16 times mutation get the best situation.

The Best Pattern : 13 -10 ,12 -10 ,-3 -16 ,11 -1 ,-11 12 ,-19 7 ,-5 -18 ,-19 -10 ,3 -11 ,14 -10 ,13 -10 ,-10 -4 ,-12 -3 ,-20 4 ,-2 -23 ,-1 3

The Best Fitness : 19

3. In the worst case, we can't find a better individual after mutation than before. Although this case is rare, it does exist.

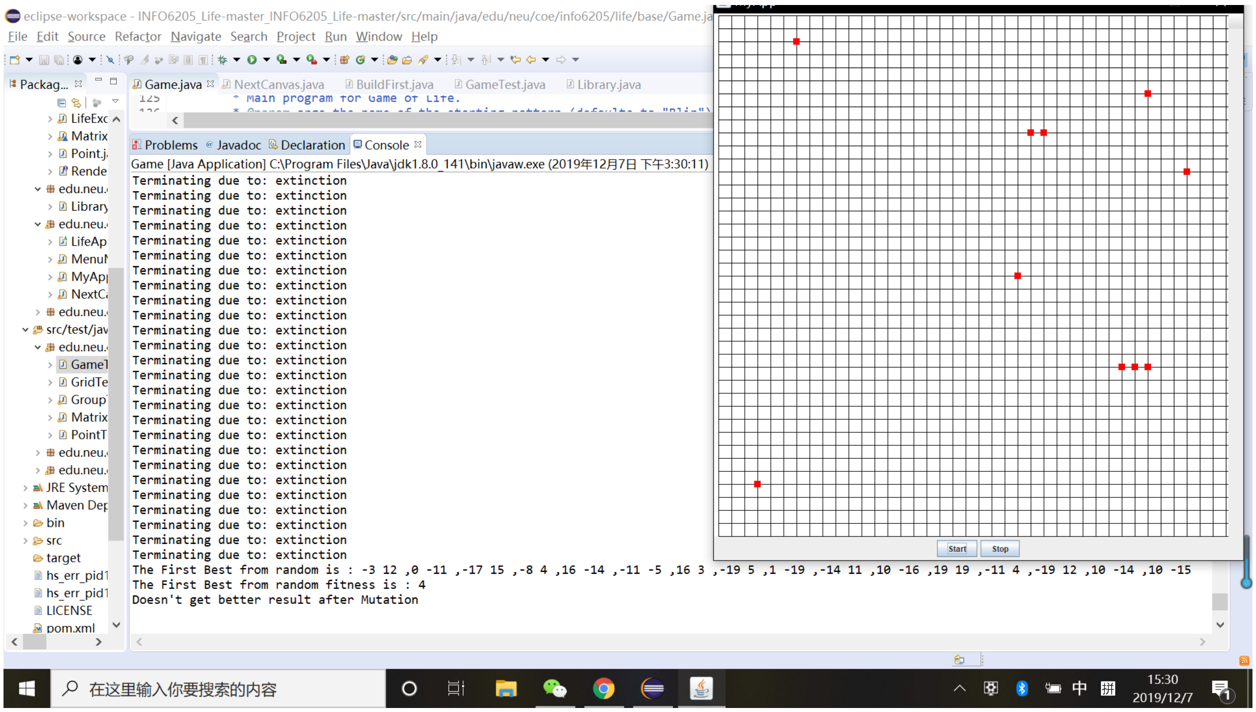

Figure 5

The First Best from random is : -3 12 ,0 -11 ,-17 15 ,-8 4 ,16 -14 ,-11 -5 ,16 3 ,-19 5 ,1 -19 ,-14 11 ,10 -16 ,19 19 ,-11 4 ,-19 12 ,10 -14 ,10 -15

The First Best from random fitness is : 4

Doesn't get better result after Mutation

7. Conclusion

After countless times selection, crossover and mutation, the 1000 final results are founded. Individual mutation will increases gene diversity and this will contributes to the richness of pattern diversity. The higher the pattern richness, the better it can adapt to the current environment. This population is more competitive in the same environment.

We generated a group of the random chromosome initially, and there is no direct relationship between members. The fitness of 10 individuals are low, meaning that it is difficult for them to evaluate over 1000 generations without any mutations. After first time mutation, better genotypes are produced, and the degree of suitability is increasing. The group begins to develop. As the fitness increases, we choose one current best pattern as gene pool to do the second mutation operation. Compared with the initial group, under most circumstances, more and more excellent individuals are produced than before.

8. Test

In our UnitTest, we tested several different aspects of our code, such as fitness function, candidate selection function, expression function and mutation function. These tests are used to verify whether the methods are reasonable and efficiency.

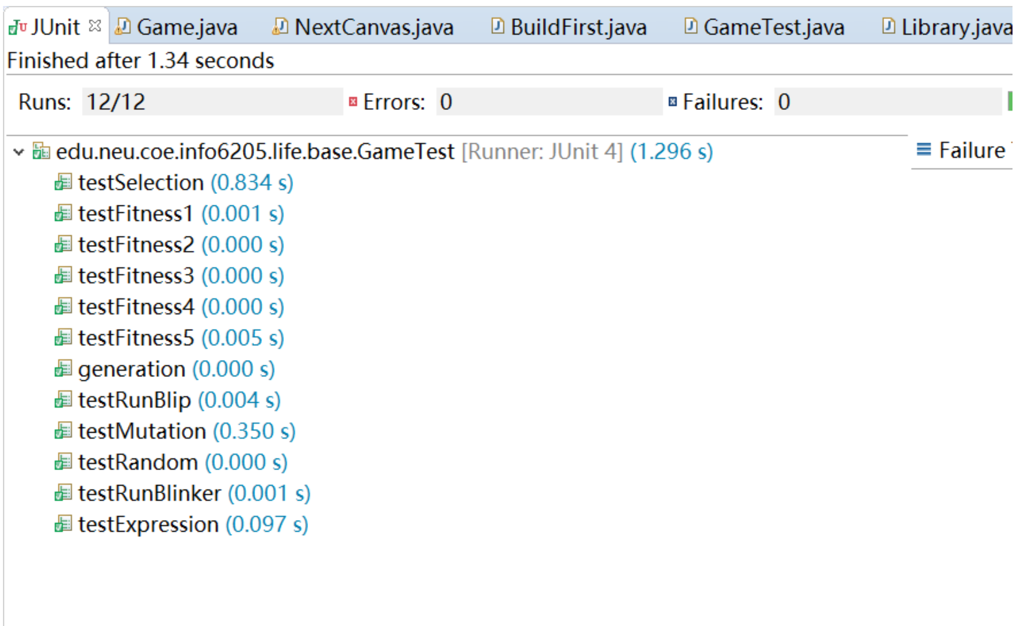

Figure 6