Report of Final Project

Problem Description:

In this project, we devote ourselves to solve *Traveling Salesman Probl*em by using Genetic Algorithm. Specifically, a delivery man needs to deliver all of the packages to different places in the city. He has to determine his initial position, and then drives to each customer' s house and, goes back to the starting point. Therefore, our team wants to help him look for the shortest route for visiting every house.

Implementation Design:

1. Genetic code: In our project, one gene represents a specific address. We consider each home as one gene, each route as a genotype(which consist of many genes). Our team has used the ArrayList to store all home and used the Array to store all routes.

2. Population

When we initialize the population, we will generate many different routes by randomly shuffle (using Collections.shuffle()) the data read from data.txt file.

3. Select candidate solutions

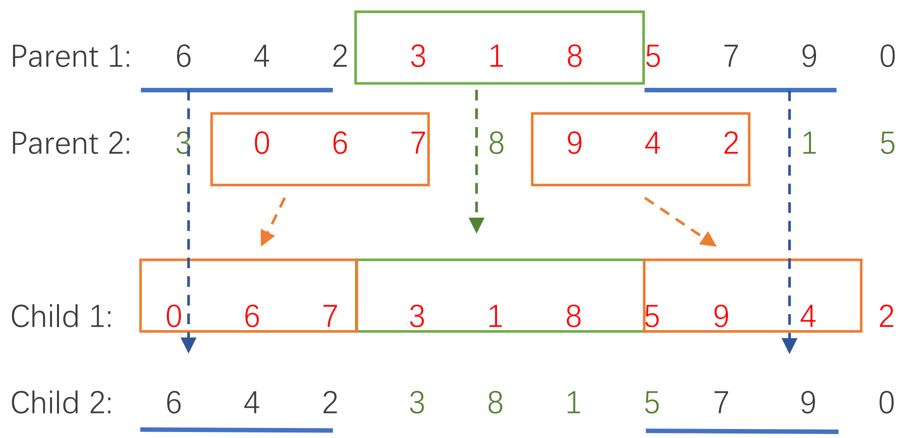
Every route has its own distance. According to the distance, we count the fitness of each route (f(xi) = 10.0 / distance(xi)).
During the process of selection, we use the Roulette Wheel Selection to choose the candidate routes from all of them. The advantage of Roulette Wheel Selection is that candidate routes with a higher fitness will be possible to be chosen. To be specific, every route has its own select probability(P(xi)) and the cumulative probability(qi). We use Math.random() to produce a random number from 0 to 1. When the random number is less than the cumulative probability of this route, the route can be selected.

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)} \qquad q_i = \sum_{j=1}^{i} P(x_j)$$

After roulette wheel selection, we will get the candidate routes for next process.

4. Crossover
   1) We randomly choose 50 pairs parents from the candidate population;
   2) Using Math.random() to produce a random number between 0 (inclusive) and 1 (exclusive);

3) If the random number is less than the Crossover Rate (0.7), parent1 and parent2 need to cross over their genes and generate two different child. Put these two children in a new population. The specific process is listed below:



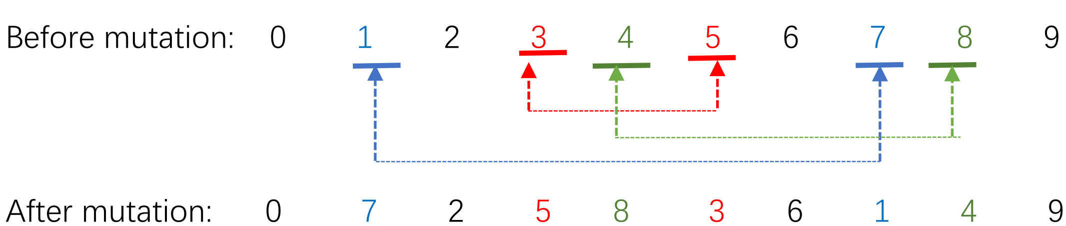
4) In other situation, if the random number is greater than the Crossover Rate (0.7), this parent does not need to cross over their genes. We just put parent1 and parent2 into the new population.

5. Sorting by Fitness

After performing crossover, we can get a new population of length 200. And then we will use Priority Queue to sort these 200 routes by fitness and get the best 100 routes.

6. Mutation

We firstly loop through every home of the route, at the same time generate a random value to check if it is less than Mutation Rate (0.015), if it is, swap this home with any other randomly chosen home in the route, otherwise, just skip this home.
The specific process is listed below:



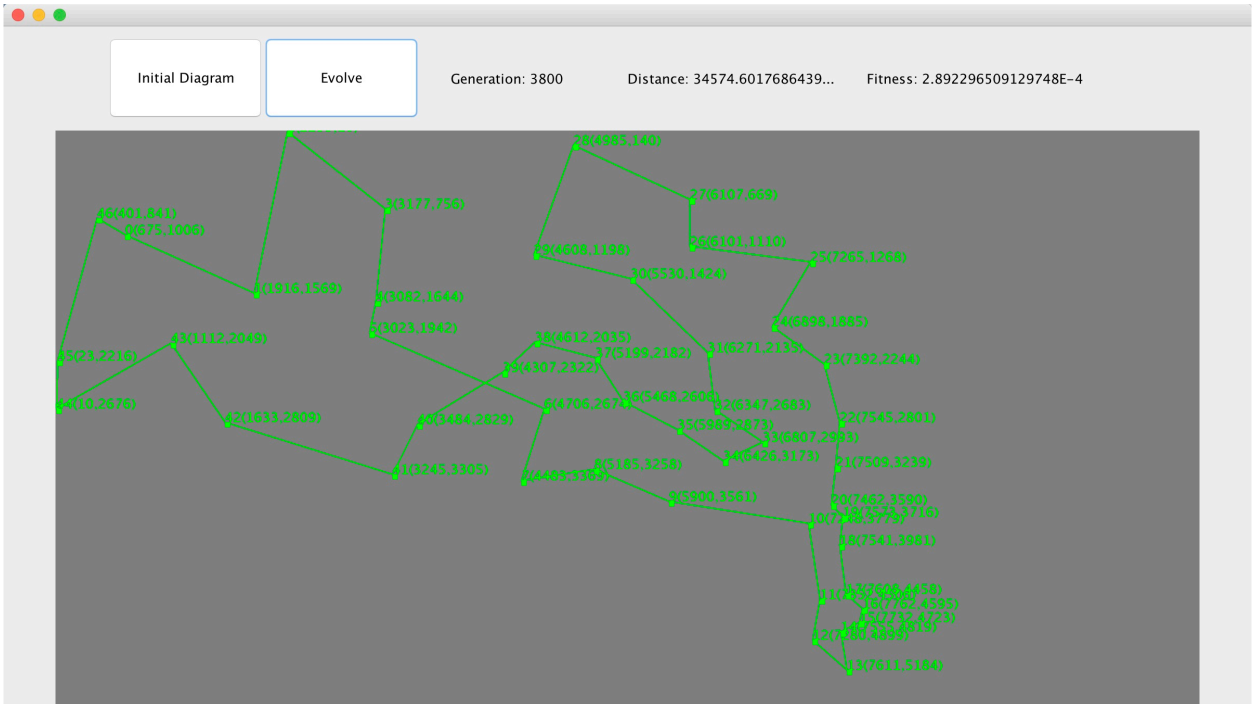
We set Mutation rate a relatively small value because we have found that if mutation rate is too big, it is very likely good results will be altered very differently.

7. User interface

In order to let the user have a vivid understanding of the progress of evolution, we have designed the user interface. The first button 'Initial Diagram' can show the best route from the initial population and its distance and fitness. Whenever we click the 'evolve' button, the population will evolve 50 times.

The running result of best route of initial population (before first evolution)

(The best route of initial population)



(The best route of the population after 3800 evolution)

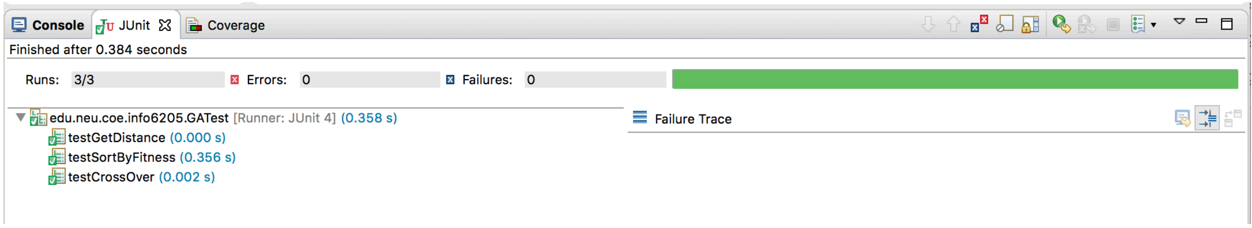
8. Logging function

We use log4j to keep track of the progress of the evolution and the best route from one population.

Results:

```
Console  JUnit  Coverage
<terminated> Main (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_141.jdk/Contents/Home/bin/java (Apr 15, 2018, 6:28:48 PM)
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  (5900, 3561) (2233, 10) (5185, 3258) (4706, 2674) (7509, 3239) (4608, 1198) (23, 2216) (7762, 4595) (1916,
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>>Distance: 165203.7956832786 || Fitness: 6.053129686663833E-5 || PI: 1.0
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is:
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  (7762, 4595) (7280, 4899) (7248, 3779) (6807, 2993) (7545, 2801) (3245, 3305) (6347, 2683) (6898, 1885) (
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 130477.04644090135 || Fitness: 7.664183297197357E-5
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  ------------------------------------------------------------------------------------------
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  After evolve:
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 10
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  (7762, 4595) (7280, 4899) (7248, 3779) (6807, 2993) (7545, 2801) (6347, 2683) (6898, 1885) (1916, 1569) (
18:28:54 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 96657.72936385153 || Fitness: 1.0345784104193786E-4
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 20
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  (7762, 4595) (7280, 4899) (7248, 3779) (6807, 2993) (7545, 2801) (6347, 2683) (6898, 1885) (7265, 1268) (
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 84840.46519722446 || Fitness: 1.1786828345122215E-4
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 30
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  (7762, 4595) (7280, 4899) (7248, 3779) (5900, 3561) (7545, 2801) (6347, 2683) (6898, 1885) (7265, 1268) (
18:28:55 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 81305.59162528452 || Fitness: 1.2299277085501444E-4
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 40
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  (7762, 4595) (7280, 4899) (7248, 3779) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (6898, 1885) (
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 77545.53316662346 || Fitness: 1.2895649293574168E-4
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 50
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  (7611, 5184) (7280, 4899) (7732, 4723) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (6898, 1885) (
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 76379.46448061315 || Fitness: 1.3092524368952898E-4
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 60
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  (7280, 4899) (7732, 4723) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (6898, 1885) (7265, 1268) (
18:28:56 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 74791.15572090463 || Fitness: 1.337056487977887E-4
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 70
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  (7611, 5184) (7280, 4899) (7732, 4723) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (7265, 1268) (
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 73077.73577607678 || Fitness: 1.368405834390078E-4
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 80
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  (7611, 5184) (7280, 4899) (7732, 4723) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (7265, 1268) (
18:28:57 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 71738.15162968611 || Fitness: 1.3939584130380465E-4
18:28:58 [main] INFO  edu.neu.coe.info6205.Main  -  >>>>>>>>>The best route is--------generation is 90
18:28:58 [main] INFO  edu.neu.coe.info6205.Main  -  (7611, 5184) (7280, 4899) (7732, 4723) (6807, 2993) (6347, 2683) (5900, 3561) (6426, 3173) (6107, 669) (7
18:28:58 [main] INFO  edu.neu.coe.info6205.Main  -  Distance: 70258.73020475042 || Fitness: 1.4233106648608158E-4
```

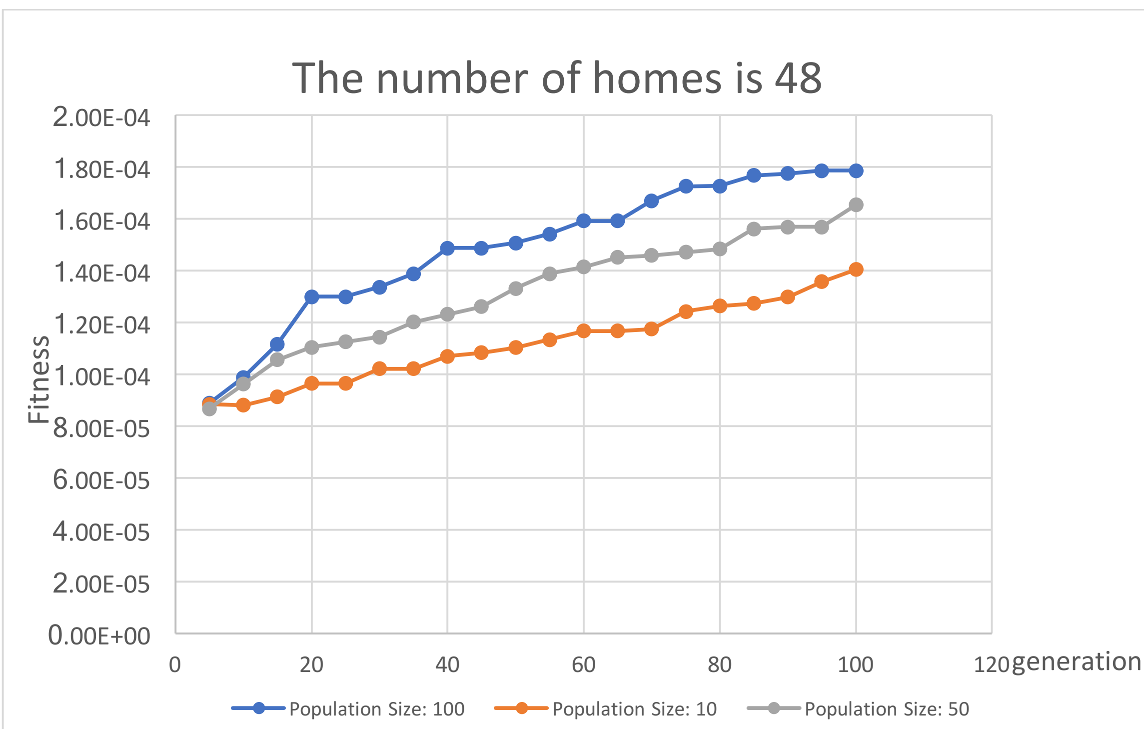
9. The result of Unit Tests:



10. Final result

Our team use 48 homes and randomly produce an initial population of length 100. After sorting the fitness of each route, we can get the initial shortest distance is about 151147. In this project, we have done experiment with 5000 times evolution. It means that we let the initial population evolve 1000 times. Finally, we can draw the conclusion that the distance of the shortest route is about 34574 in about 3800th generation.

Conclusions:



The above graph shows that more routes in one population we have, much greater fitness of shortest route we can achieve with less iterations. It is because that if there are a lot of routes in one population more routes will be likely to cross over and mutate and then generate more possibilities and find the best. In this way, we can easily get the better solution quicker.

Also, after several times experiments we have found that the best fitness from the population will become stable after several times evolution. However, the best route of each population will not become worse because we have saved the best route from the last population.