Genetic Algorithm: The Location Selection Problem

Kun Du, Wenyin Zhao

I: Problem

We want to set up a vegetable transfer station between vegetable base A and cities: B, C, D, E, F, G, H, I, G, K. The transfer station should meet the following requirements as much as possible:

（1） In order to save transportation costs, the distance between the transfer station and the vegetable base A and various locations should be as small as possible

（2） In order to save the cost of construction fee, the transfer station should be as closer to the vegetable base as possible, because the closer the transfer station is to the city, the higher will construction cost rise.

II: Genetic Algorithm

1. *Genetic code*

Each transfer station consists of a coordinate ***(x, y).*** In our project, we use binary coding to express to convert *x* and *y*. The code length is 20, the first 10 bits are ***x,*** and the last 10 bits are ***y***. For example, if there is a transfer station's coordinate is (20, 50), its gene would be:

0000010100|0001100010

Thus:

Phenotype:

Each chromosome (transit station) is represented by a set of coordinates, such as (20,50)

Genotype:

A set of 20 bits binary codes, for example: 00000101000001100010

2. *Fitness function (Evaluation function)*

The fitness of each chromosome (transfer station) is composed of two parts. The first part is the distance sum between the transfer station and the vegetable base A. In order to save transportation costs, the total distance should be as small as possible; the second part is the transfer station. The second part is the distance of the transfer station from the vegetable base A. And we want the transfer station should be as close to vegetable base A as possible, because the closer to cities, the rental cost and construction fee would be higher.

We assigned each part a corresponding weight ***w.*** The first part's proportion is ***0.7*** and the second part is ***0.3.*** So the optimal transfer station construction site should have the minimum sum of fitness from two parts. We used the following function to calculate the fitness.

$$Fitness(x)=\sum_{i=0}^{10}\left(\sqrt{(x-x_i)^2+(y-y_i)^2}\right)\cdot W_1+\sqrt{(x-x_0)^2+(y-y_0)^2}\cdot W_2$$

For analysis convenience, we divide the fitness of each chromosome by 100

$$Fitness(x)=\frac{\left(\sum_{i=0}^{10}\left(\sqrt{(x-x_i)^2+(y-y_i)^2}\right)\cdot W_1+\sqrt{(x-x_0)^2+(y-y_0)^2}\cdot W_2\right)}{100}$$

3. *Initial population*

In the beginning, we randomly generated 30 coordinates (chromosomes) to form the initial population and evolved on this basis.

4.Evolution

For each generation of chromosomes, the best chromosomes are selected first; the cumulative probability of chromosomes is calculated to provide a basis for selecting other chromosomes as parents for the next generation; then we cross and mutate the selected chromosomes.

5. *Natural Selection*

Best Chromosomes Selection:

When selecting the next generation's parent chromosomes, we use *ProrityQueue* to sort the chromosomes according to the fitness of each chromosome (Our chromosomes class implement the comparable interface and override compareTo method), select the best chromosomes, and copy them directly to the next generation ***without*** crossing over and mutation.

Other Parent Chromosomes Selection:

We used the "roulette method" when selecting other parents' chromosomes.

Step (1) We calculate the probability that each chromosome will be selected for each generation based on the fitness of each chromosome. Since we want the better the chromosome's fitness is, the larger possible it be selected, we divide each Chromosome by 10 (10/fitness). For example: if the population size is 5, the fitness of each chromosome is (10, 20, 30, 40, 50). The new fitness after calculation of each chromosome is (1, 0.5, 0.3, 0.25, 0.2)

Step (2) Calculate the selected probability (fitness/SumFitness(2.5)) of each chromosome in the population. Using the above example, the probability of being selected in each chromosome is (0.44.22, 0.14, 0.12, 0.08).

Step (3) Calculate the cumulative probability that each chromosome is selected in the population, using the above example, (0.44,0.66,0.8,0.92,1.0)

Step (4) The computer generates a random probability. If the probability is less than a certain cumulative probability, the corresponding chromosome is selected. And the better the chromosome will be easier to be selected.

6.Crossing Over:

After we selected parent chromosomes, if the random generated probability is less than Pc(0.8) we randomly exchange fragments in parent chromosomes to cross over.

For example:

Parent c1 ： 00|1101|10010011001100

Parent c2 ： 10|1001|01000000100101

Swap gene fragment index 2-5:

Child 1 ： 00|1001|10010011001100

Child 2 ： 10|1101|01000000100101

Copy chromosomes to the next generation and repeat the process for Parent2.

5. *Mutation:*

If the random probability is less than the probability of variation Pm(0.1)， chromosomes will mutate. The mutation is a random change on one gene of a chromosomal， and mutate to 0(if the gene was 1) or 1(if the gene was 0).
For example:
Chromosome c gene is：
00110110010011001100
The mutation randomly mutated gene on index 4, and the gene after mutation is：
00111110010011001100

6:Records of evolution process

In the process of evolution, we will record the best chromosomes, the generation when the best chromosomes shows and the best chromosomes of each generation. Specific evolutionary process is recorded in the log file named log.txt. The following screenshot is some of the records.

```
08:18:57 [ main ] Build GeneticAlgorithm
08:18:57 [ main ] Initialize distance matrix
08:18:57 [ main ] Initialize the population. The population is: 30
08:18:57 [ main ] info of each Chromosome from init Population
08:18:57 [ main ] x: 904 y: 292
08:18:57 [ main ] The sum distance to each city: 6670.8027 Distance to first city: 916.52606 Chromosome fitness is:  49.4452
08:18:57 [ main ] x: 287 y: 351
08:18:57 [ main ] The sum distance to each city: 4339.253 Distance to first city: 402.35556 Chromosome fitness is:  31.581839
08:18:57 [ main ] x: 855 y: 256
08:18:57 [ main ] The sum distance to each city: 6259.487 Distance to first city: 860.03546 Chromosome fitness is:  46.396515
08:18:57 [ main ] x: 534 y: 370
08:18:57 [ main ] The sum distance to each city: 4496.8276 Distance to first city: 605.47174 Chromosome fitness is:  33.29421
08:18:57 [ main ] x: 617 y: 660
08:18:57 [ main ] The sum distance to each city: 6019.447 Distance to first city: 853.5274 Chromosome fitness is:  44.69671
08:18:57 [ main ] x: 164 y: 165
08:18:57 [ main ] The sum distance to each city: 4494.891 Distance to first city: 184.28511 Chromosome fitness is:  32.017094
08:18:57 [ main ] x: 597 y: 208
08:18:57 [ main ] The sum distance to each city: 4621.579 Distance to first city: 598.2416 Chromosome fitness is:  34.145775
08:18:57 [ main ] x: 689 y: 369
08:18:57 [ main ] The sum distance to each city: 5230.1606 Distance to first city: 741.1626 Chromosome fitness is:  38.83461
08:18:57 [ main ] x: 490 y: 101
08:18:57 [ main ] The sum distance to each city: 4565.0273 Distance to first city: 472.7589 Chromosome fitness is:  33.373466
08:18:57 [ main ] x: 51 y: 752
08:18:57 [ main ] The sum distance to each city: 7541.274 Distance to first city: 702.68414 Chromosome fitness is:  54.89697
08:18:57 [ main ] x: 649 y: 518
08:18:57 [ main ] The sum distance to each city: 5475.5195 Distance to first city: 784.00574 Chromosome fitness is:  40.680653
08:18:57 [ main ] x: 822 y: 64
08:18:57 [ main ] The sum distance to each city: 6326.529 Distance to first city: 802.1222 Chromosome fitness is:  46.69207
08:18:57 [ main ] x: 26 y: 541
08:18:57 [ main ] The sum distance to each city: 6457.8213 Distance to first city: 491.03665 Chromosome fitness is:  46.677856
08:18:57 [ main ] x: 155 y: 53
```

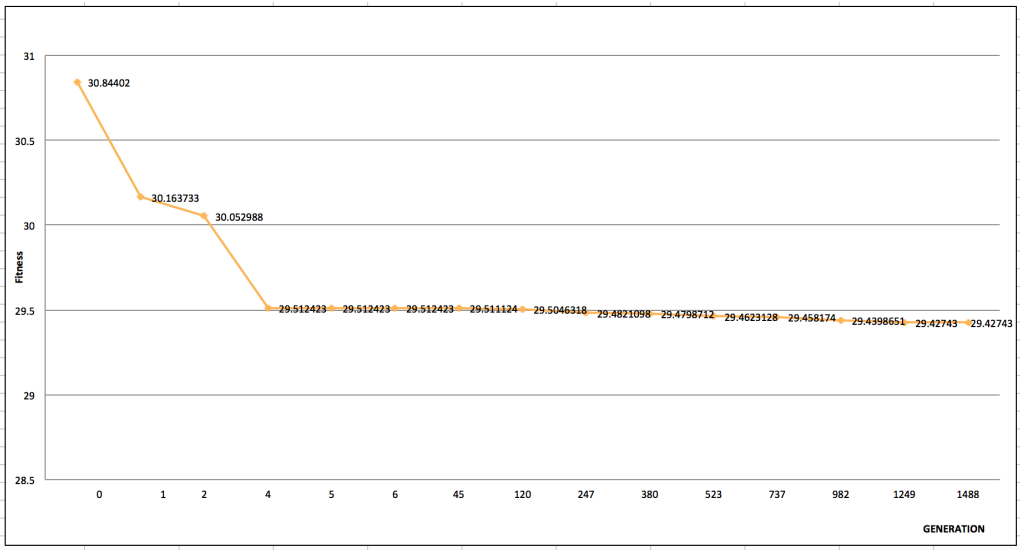
III: Results
We ran on 1500 generations to get the best chromosome and its generation

```
run:
From these 1500 generation we can get the Result as follows:

The Generation of the most fit chromosome generates at: 1393
The most fit x is: 348
The most fit y is: 261
Best fitness value is ：29.420637
The distance sum to each city is(km): 4035.8025
The distance to Vegetable Base is(km): 390.0064
成功构建（总时间：2 秒）
```

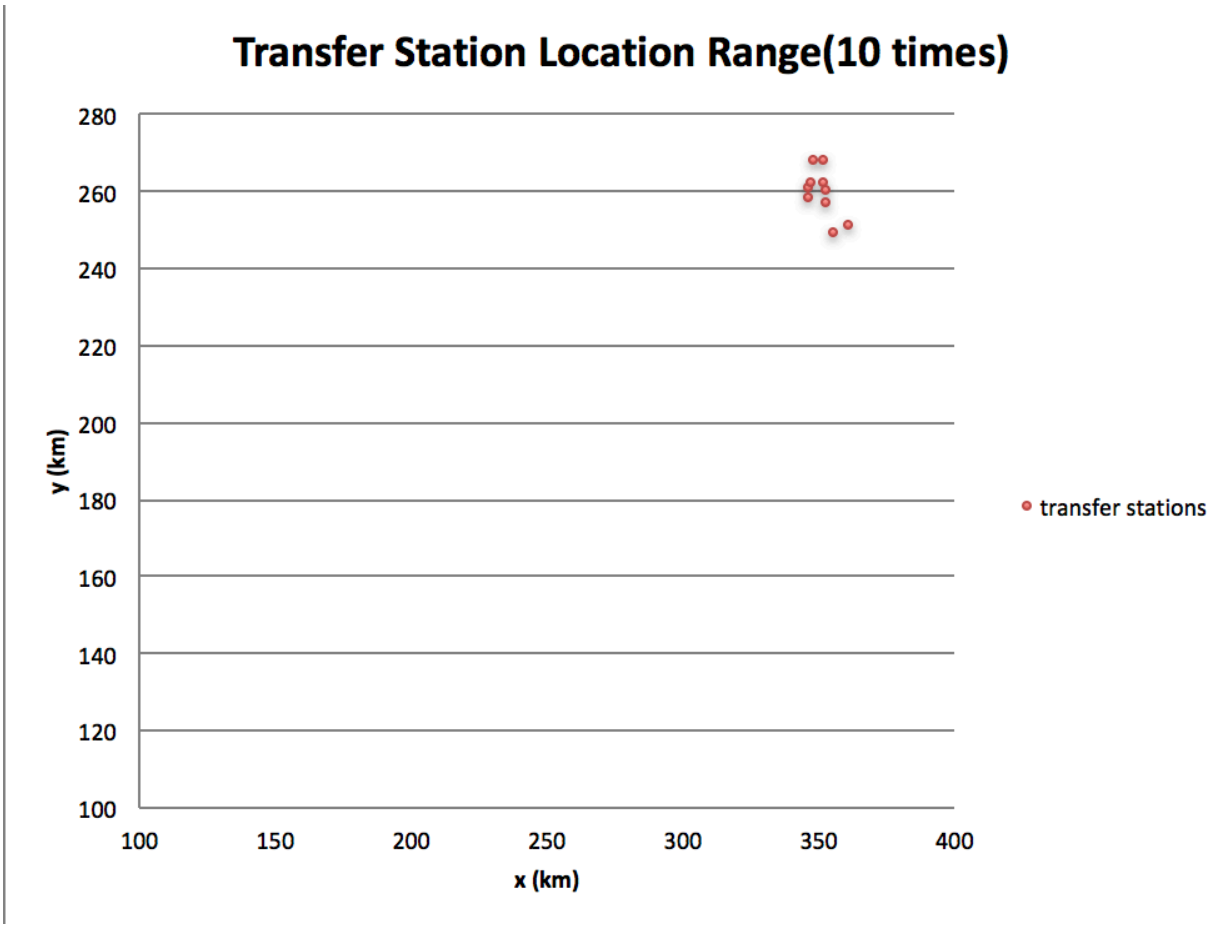
The following graph is best chromosome's fitness of each generation: we can see the value is growing to more and more stable, and the best chromosome fitness is around 29.42

0	30.84402
1	30.163733
2	30.052988
4	29.512423
5	29.512423
6	29.512423
45	29.511124
120	29.5046318
247	29.4821098
380	29.4798712
523	29.4623128
737	29.458174
982	29.4398651
1249	29.42743
1488	29.42743

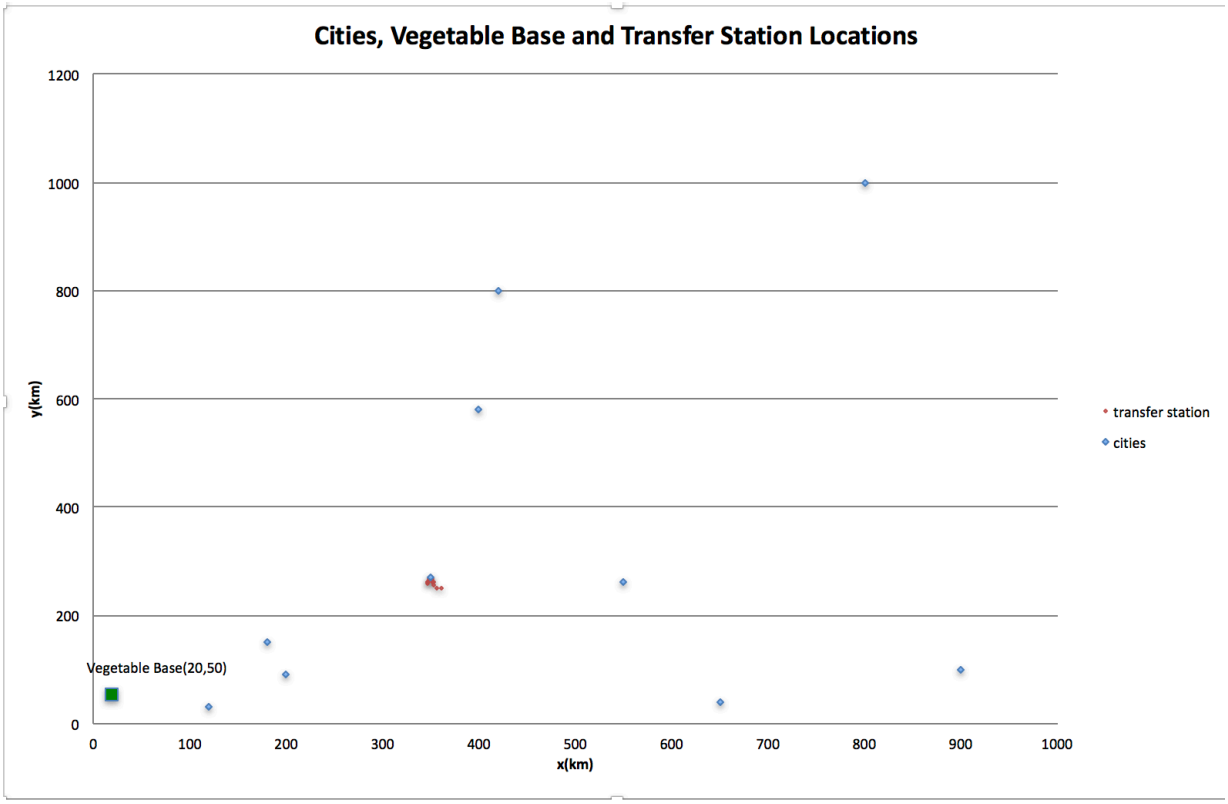


Location Selection:

After running the program 10 times, we got the 10 best chromosomes and mapped the position of each of the best chromosomes (transfer stations) in excel. The results are shown in the figure:



This is the location relationship amomg transfer station, vegetable base A and cities



Unit Tests passed screenshot

We tested 6 main functions and all passed