

Genetic Algorithm: Maze Problem

Team No. 201

Ruixing Li 001886987

Chen Qiu 001877500

Problem Description

Generally, Given a maze, indicating the starting point and the ending point, finding a valid and feasible path from the starting point to the ending point is a maze problem.

In our project, we make a little change from maze problem, the player is supposed to find a gem in the maze first, use the gem to skip the trap and find a way to the exit.

GA Components

GA code typically has the following functions:

1.Genetic code

Initially, we set 100 players in maze and the maximum number of generation is 30000. A player has 200 genes which means that he/she can move 200 steps at most, a gene is generated randomly from parent.

2.Fitness Function.

Probably the most important bit of a GA next to figuring out how to encode the population's genes. Well, in our project, genes represent the move direction down, left, up, right encoded as 1,2,3,4 in the maze

We applied BFS to get the shortest distance as well as the path among gem, trap, and exit position, if a player is able to collect more gems, correspondingly he/she will get more score in the process.

Additionally, if any of the moves go over 200 steps, we impose a penalty on the score. Which mean that the player should end the game and lose all score right away.

3.Survival Function.

We applied priorityqueue and rewrite the compare function to get the top 5 highest score which we calculate from fitness function for the generation, these candidates will play the role as parents to get offspring

4.Crossover.

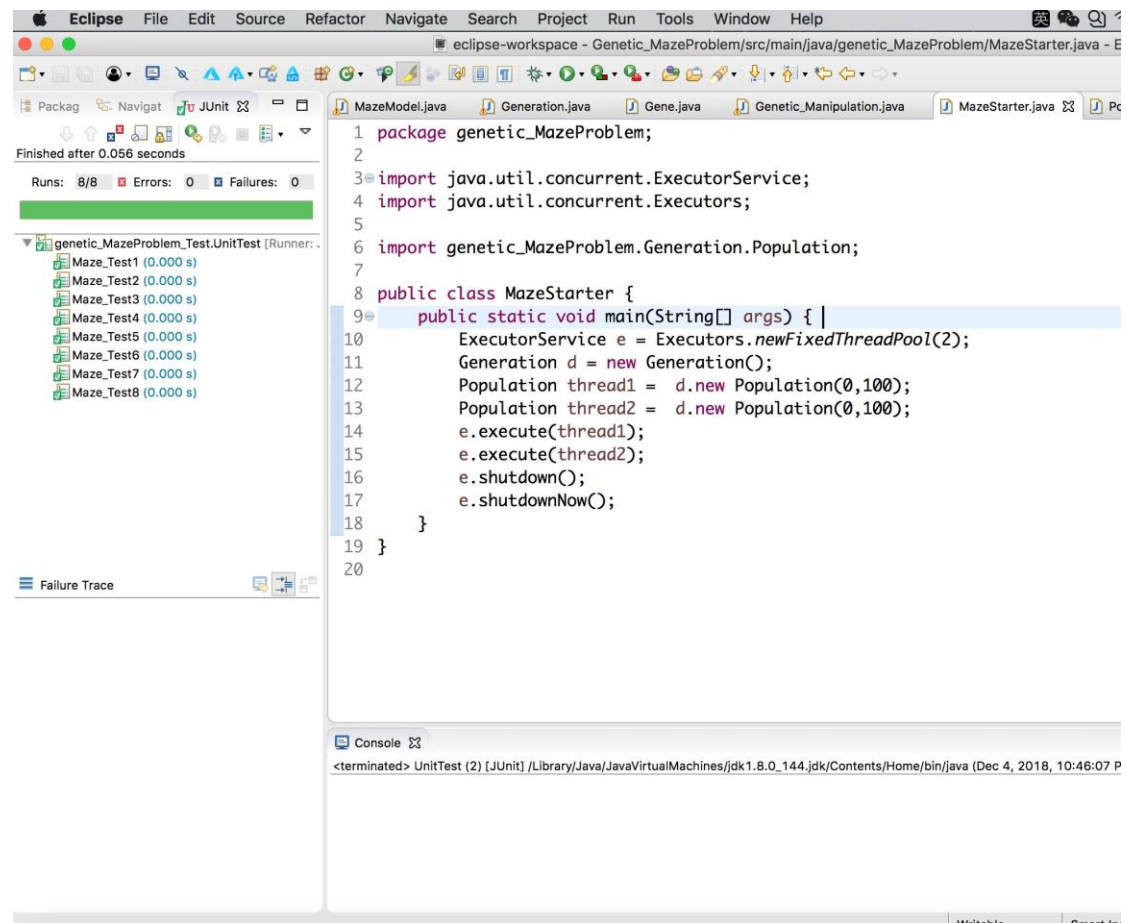
For each generation, we select the top 5 players who get the highest scores as parent to generate offspring, to get enough offspring, we do crossover multiple times

5.Mutation.

Some of the genes have a random chance of mutating when producing offspring. Mutation plays a pretty important part of the GA. If there is no mutation then the solution will kind of all end up being the same with no chance of breaking free for other potential solutions. In our project, we set

mutation rate to 0.005, which the probability to mutation for each gene is 0.5%. So roughly 91% of the time we should have a mutation occurring for a player.

4. Unit Tests



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Tools, Window, and Help. The toolbar contains various icons for file operations and development tools. The left sidebar shows the Package Explorer with a tree view of the project structure. The main editor window displays the source code for MazeStarter.java. The code is as follows:

```
1 package genetic_MazeProblem;
2
3 import java.util.concurrent.ExecutorService;
4 import java.util.concurrent.Executors;
5
6 import genetic_MazeProblem.Generation.Population;
7
8 public class MazeStarter {
9     public static void main(String[] args) {
10         ExecutorService e = Executors.newFixedThreadPool(2);
11         Generation d = new Generation();
12         Population thread1 = d.new Population(0,100);
13         Population thread2 = d.new Population(0,100);
14         e.execute(thread1);
15         e.execute(thread2);
16         e.shutdown();
17         e.shutdownNow();
18     }
19 }
20
```

The bottom of the IDE shows the Console window with the following output:

```
<terminated> UnitTest (2) [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Dec 4, 2018, 10:46:07 P
```

5. Parallel processing

We conducted two threads simulating two population of same creature with base amount of 100 each. And merge the result together in priorityqueue to get parents.

6. Conclusion

From the screen shot we can see that finally the play's status turns to 3

which means that he/she get to the exit successfully, you can see the thread state and generation state from the console.

```
1 package genetic_MazeProblem;
2
3 import java.util.concurrent.ExecutorService;
4
5
6
7
8 public class MazeStarter {
9     public static void main(String[] args) {
10         ExecutorService e = Executors.newFixedThreadPool(2);
11         Generation d = new Generation();
12         Population thread1 = d.new Population(0,100);
13         Population thread2 = d.new Population(0,100);
14         e.execute(thread1);
15         e.execute(thread2);
16         e.shutdown();
17     }
18 }
```

Console Output:

```
<terminated> MazeStarter (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Dec 4, 2018, 11:17:52 PM)
Current Thread: 11 Current Thread: 10 Generation 21 Position: 8,17 Status: -1 Score: 500.00 FoundTrap: false hasGems: false
Generation 21 Position: 8,17 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 11 Generation 31 Position: 7,16 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 10 Generation 41 Position: 7,16 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 11 Generation 51 Position: 2, 11 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 10 Generation 61 Position: 1, 11 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 11 Generation 71 Position: 1, 51 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 10 Generation 81 Position: 7, 41 Status: -1 Score: 500.00 FoundTrap: false hasGems: false passTrap: false
Current Thread: 11 Generation 91 Position: 2, 51 Status: -1 Score: 500.00 FoundTrap: false hasGems: true passTrap: false
Current Thread: 10 Generation 101 Position: 3, 51 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 111 Position: 7,16 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 121 Position: 11,16 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 131 Position: 3,17 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 141 Position: 8,17 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 151 Position: 2, 51 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 161 Position: 3, 51 Status: -1 Score: 3250.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 171 Position: 1, 51 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 181 Position: 3, 51 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 191 Position: 8,15 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 231 Position: 1, 91 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 241 Position: 4, 41 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Current Thread: 10 Generation 261 Position: 17, 31 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Generation 261 Position: 17, 31 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 271 Position: 7,10 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 281 Position: 7,10 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 291 Position: 3,17 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 301 Position: 8,15 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 311 Position: 17, 11 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 321 Position: 1, 51 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 331 Position: 1, 31 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 341 Position: 8, 41 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 351 Position: 3, 11 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 361 Position: 1, 71 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 371 Position: 2, 71 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 10 Generation 381 Position: 1, 31 Status: -1 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
Current Thread: 11 Generation 391 Position: 17, 31 Status: 31 Score: 4000.00 FoundTrap: true hasGems: true passTrap: true
```