

## Some Implementation Details in GA.cpp

### 1 Necessary Parameters

The appropriate population size should be in  $[20, 200]$ . I choose  $100$ .

The appropriate probability of crossover should be in  $[0.4, 0.99]$ . I choose  $0.7$ .

The appropriate probability of mutation should be in  $[0.005, 0.01]$ . I choose  $0.07$ .

The appropriate maximum number of generations (one of the stopping criterions) should be in  $[100, 1000]$ . I choose  $500$ .

### 2 Encoding Method

To turn each  $x$  into a 16-bit binary digit, namely turn  $[-1, 15]$  into  $[0, 1111\ 1111\ 1111\ 1111]$ ,

I do

$$xdna = BIN(2^{(x+1)} - 1).$$

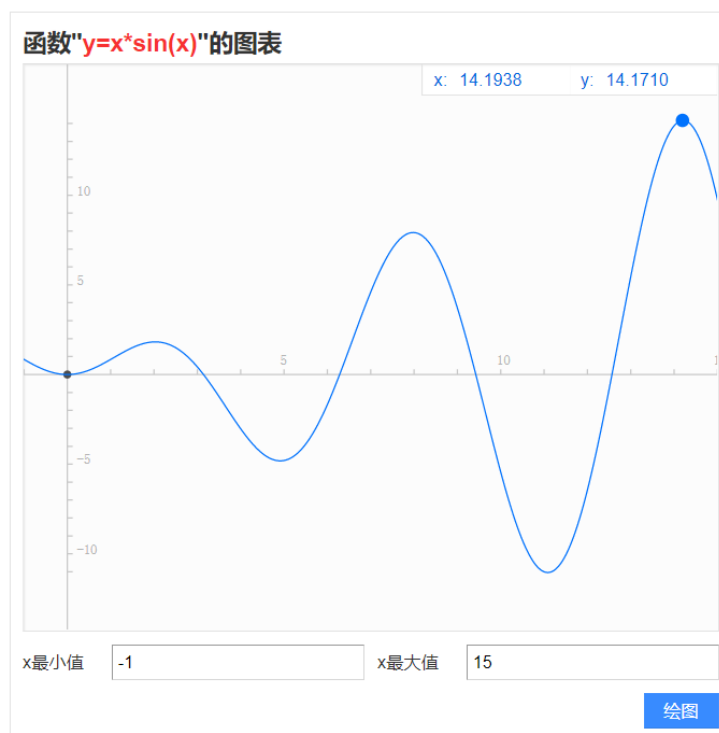
### 3 Fitness Score

Firstly, we need to obtain the  $x$  which can make  $f(x)$  smallest.

Firstly, we hope that the fitness score is the highest when the  $x$  is optimal.

Secondly, because we need to use fitness scores to calculate the select-rates, we have to make the fitness scores all positive.

Then I use *baidu* to draw the image of  $f(x) = x \cdot \sin(x)$ .



We can see the max value of  $f(x)$  is smaller than 15.

Above all, I do

$$fitness = -x * \sin(x) + 15$$

to get legal fitness scores.

#### 4 Selection Method

I choose a roulette-like way to select offspring.

Get the probability through

$$P(x_i) = \frac{fitness(x_i)}{\sum_{i=0}^n fitness(x_i)}.$$

The  $x$  which has higher fitness score will be selected more probably.

#### 5 Crossover Method

I choose single-point crossover method.

Between each pair of parents in population, crossover will happen at a probability of 70%. If the crossover happens, we randomly choose a position  $i$ , then exchange the front  $i$  digits of DNA of  $x_{2i}$  and  $x_{2i+1}$ .

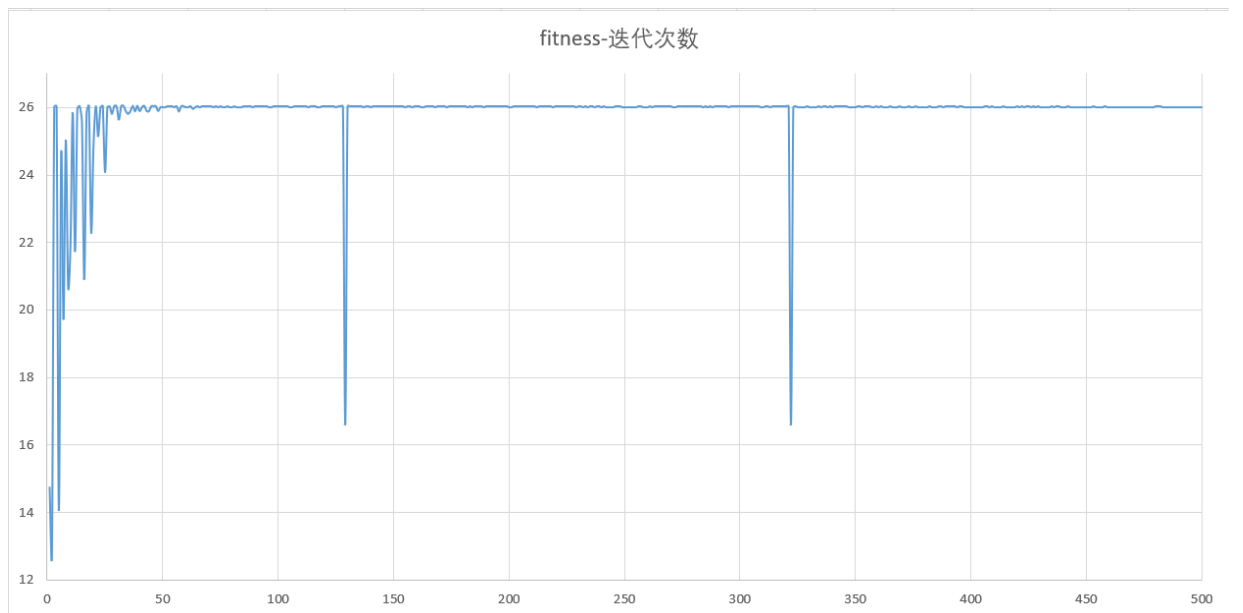
#### 6 Mutation Method

I choose bit mutation method.

Every bit in DNA of each  $x$  will mutate at a probability of 7%. If the mutation happens, original  $0$  will change to  $1$ , original  $1$  will change to  $0$ .

#### 7 Experimental Result

```
Microsoft Visual Studio 调试控制台
In 477 generation: x* = 11.0004 with fitness score 26.0002
In 478 generation: x* = 11.0004 with fitness score 26.0002
In 479 generation: x* = 11.0004 with fitness score 26.0002
In 480 generation: x* = 11.0447 with fitness score 26.0314
In 481 generation: x* = 11.0447 with fitness score 26.0314
In 482 generation: x* = 11.0447 with fitness score 26.0314
In 483 generation: x* = 11.0004 with fitness score 26.0002
In 484 generation: x* = 11.0004 with fitness score 26.0002
In 485 generation: x* = 11.0004 with fitness score 26.0002
In 486 generation: x* = 11.0004 with fitness score 26.0002
In 487 generation: x* = 11.0004 with fitness score 26.0002
In 488 generation: x* = 11.0004 with fitness score 26.0002
In 489 generation: x* = 11.0004 with fitness score 26.0002
In 490 generation: x* = 11.0004 with fitness score 26.0002
In 491 generation: x* = 11.0004 with fitness score 26.0002
In 492 generation: x* = 11.0004 with fitness score 26.0002
In 493 generation: x* = 11.0004 with fitness score 26.0002
In 494 generation: x* = 11.0004 with fitness score 26.0002
In 495 generation: x* = 11.0004 with fitness score 26.0002
In 496 generation: x* = 11.0004 with fitness score 26.0002
In 497 generation: x* = 11.0004 with fitness score 26.0002
In 498 generation: x* = 11.0004 with fitness score 26.0002
In 499 generation: x* = 11.0004 with fitness score 26.0002
In 500 generation: x* = 11.0004 with fitness score 26.0002
D:\个人文件夹\... \大二上\问题求解与实践\Homework7\Genetic Algorithms\Debug\Genetic Algorithms.exe (进程 13912) 已退出
。 代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . .
```



The result I obtained through GA.cpp is

$$x^* = 11.0004$$

The relative error is

$$\eta = (11.0857 - 11.0004) / 11.0857 * 100\% = 0.77\%$$