

# 遗传算法实现报告

王资 18214668 计算机技术

## 1.基本思想

利用遗传学的思想，对种群的个体进行优化选择，对留下的个体进行染色体交换，以及存在的基因突变，这样产生下一代，然后重复这个操作，最终将得到一个适应性较高的种群，这个种群中适应性最高的染色体通过解码后作为最优的解。

## 2.代码中的数据结构

### 2.1 曲线 $x\sin(7/x)+3$ 最高值

`vector<int> group` 代表种群，该容器的每个值代表个体的染色体；`vector<int> choose` 存活下来的个体，下标与 `group` 的一致；`vector<double> fs` 各个个体的适应度，通过  $x\sin(7/x)+3$  计算得出。

### 2.2 n 后问题

与上面类似，`vector<vector<int>> group` 代表种群，该容器的每个值代表个体的染色体，即棋盘；`vector<int> choose` 存活下来的个体，下标与 `group` 的一致；`vector<double> fs` 各个个体的适应度，通过计算棋盘中冲突皇后数得出，由于遗传算法的最终目标是得到最大值，因此适应度等于冲突皇后数对棋盘尺寸取补。

## 3.程序流程

求曲线  $x\sin(7/x)+3$  最高值和 n 后问题的主体流程是一致的，只有染色体交叉和基因突变的策略有所不同，以及 n 后问题在迭代过程中有解时即可中止，而求曲线最大值需要迭代到最后。

- 1) 随机产生第 0 代种群的个体。
- 2) 迭代下述过程，直到达到最大迭代次数
- 3) 计算这些个体的适应度，并根据适应度加权，选择可以存活下来的个体。
- 4) 若符合染色体交叉的概率，存活的个体进行染色体交叉产生后代，实现中使用简单的相邻两个个体交配。对二进制编码的染色体可以随机选择一个位置，然后交换该位置右边的基因；而对于排列类问题，可以随机选择一个位置，选择该位置后面的基因，然后在另一个染色体上找到这些基因，按顺序替换成第一个染色体选择了出来的基因。

- 5) 若符合基因突变的概率，对产生的后代进行基因突变。对二进制编码的染色体可以对某一位取反，对于排列类问题，可以随机交换两个基因的位置。
- 6) 若找到最优解，跳出循环，否则转回 2)
- 7) 输出最优解，即当前种群中适应值最高的个体的染色体。

## 4.调试过程出现的问题和解决方法

- 1、对于选择存活对象的实现，先取整数部分，再将小数部分由大到小排序，选择前 k 个，k 为种群个体数剩余名额，这个实现需要根据数组各个 offset 对应的值对数组的下标排序而非原数组排序。搜索资料后找到了可以实现这种方法的方法。
- 2、二进制的操作可以直接使用位操作完成，而不需要创建较长的数组，并且使用位运算，可以简单的得到某个位置上右边的高位和左边的低位，对某位取反也能简单实现，更重要的是求适应值的时候可以直接映射。
- 3、n 后问题不是使用二进制编码的，因此染色体交叉的方法需要使用另一种方法实现，我使用的方法是随机选择一个位置，选择该位置后面的基因，然后在另一个染色体上找到这些基因，按顺序替换成第一个染色体选择了出来的基因，如对于两组染色体 0 1 2 3 4，4 2 1 0 3，交换后三位的结果是 0 1 4 2 3、2 3 1 0 4，即交叉到另一染色体的基因会保持原来的顺序，这样的交换也能保证每个个体都是合法的。基因突变的修改方法较简单，可以改成交换两个基因的位置，保证个体是合法的。

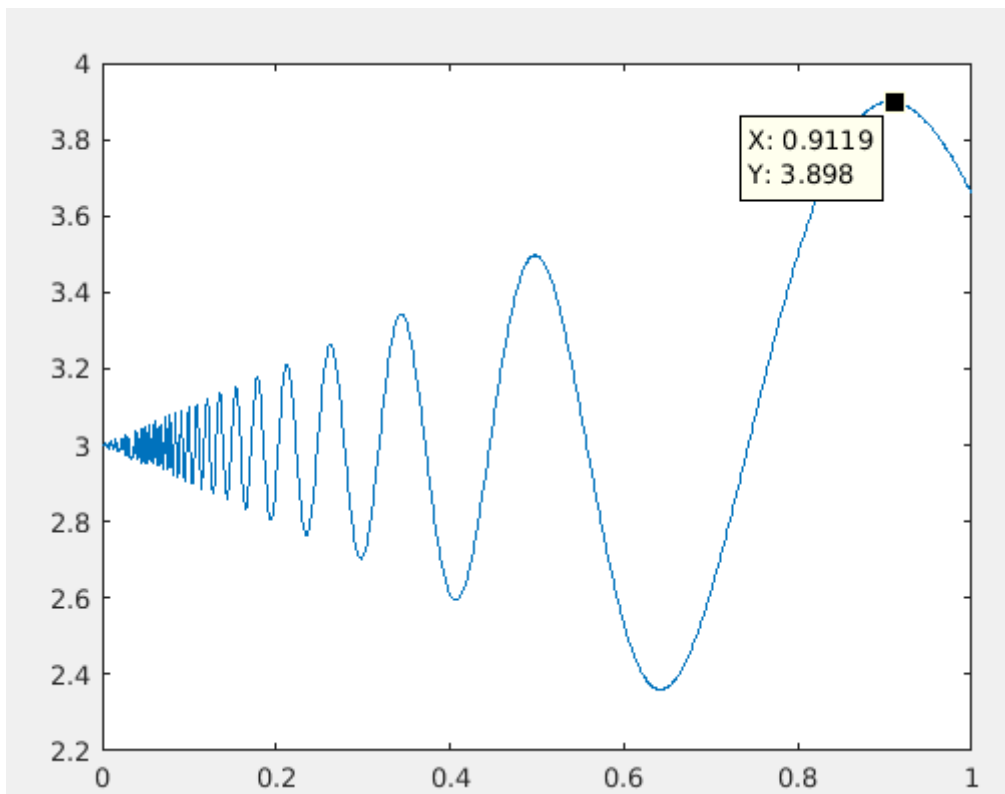
## 5.运行结果

### 5.1 求曲线 $x\sin(7/x)+3$ 最高值

```
/home/wz/CLionProjects/GA/cmake-build-debug/GA
The best code is 935 which is also float 0.913978 and the point 3.89663

Process finished with exit code 0
```

使用 MATLAB 得到对应曲线



可以看出最大值在 0.911 附近，最大值为 3.898，与得到的结果相近，产生误差的原因是为了保证  $x\sin(7/x)+3$  不会产生上溢，在分母位置加了一个很小的数。

## 5.2n 后问题

```
/home/wz/CLionProjects/n-queens/cmake-build-debug/n_queens
60 41 19 89 34 58 73 20 42 16 92 83 40 23 99 47 77 32 15 95 64 56 46 49 29 71 8 93 11 30 39 6 51 97 55 21 91 74 36 78 85 2 33 35 62 86 54 98 90 38 72 82 18
100-queens in genetic algorithm takes 91s
Process finished with exit code 0
```

遗传算法可以找到解，但速度不如模拟退火。

## 6.总结

本次实验实现了遗传算法，包括最直接的应用，求一个函数的最大值，和应用在 n 后问题上。对于可以使用二进制编码的问题，需要运用好位操作的方法对染色体进行交叉、基因突变，同时要准确地建立映射关系，即解码。而对于 n 后问题这类排列型的问题，对于一般的遗传算法改动会较大，染色体交叉和基因突变的方法要改变，否则大部分的个体都是不合法的。

遗传算法有很多参数，包括最大迭代次数、每代种群个体数目、交叉概率、突变概率等，而且交叉和突变的方法也可以有各种定义的方式，所以虽然实现这次的实验只调试了少部分参数，使用的交叉和突变算法也有限，但在实现过程中就可以想象得到遗传算法的扩展是多种多样的。

## 7.存在问题和改进设想

n 后问题中为了实现上述的较复杂的染色体交换，需要遍历染色体（长度为  $n$ ）以及截取的位置  $k$ （长度为  $n-k$ ）也即每次染色体交换的时间复杂度是  $O(n^2)$ ，这样的低效率大大影响了算法的效率，但如果使用一般的交叉方式，可以预想到大部分的个体都是非法而被抛弃，最终需要交配更多的个体才能得到个体数量为  $m$  的种群。因此结论是想 n 后问题这种排列类问题不适合使用遗传算法，因为染色体交叉的操作代价太大，在网上搜索相关问题时，也得到了相似的反馈，即使用遗传算法解 n 后问题的效率普遍低下。若存在改进的方法，那就是找到一个比较高效的染色体交叉的方法了。