统计推断在数模转换系统中的应用

组号：26　小组成员：王天祎 5140309214　李琰 5140309225

摘要：统计推断是数模转换系统中应用的重要方法。本课题主要内容是检测某型投入批量试生产电子产品模块内输入输出特性呈明显非线性的传感器部件，对其设计了一种成本合理的定标校准方案，并确定其输入输出信号之间的关系。我们使用 MATLAB 进行设计，采用遗传算法选择合理的定标点，最后使用插值或拟合的方法来完成定标与校准工作。

关键词： 统计推断，定标，三次样条插值法，遗传算法，多项式拟合

Statistical Inference for the data of Digital-analog conversion system

Abstract: Statistical Inference is an important application in the data of digital-analog conversion system. The main content of this topic is monitoring and test a sensor component which has a obviously nonlinear input-output characteristic in a type of electronic products which is put into production. We have designed a scaling calibration scheme with a reasonable cost to find the relationship between input and output signals. We use MATLAB to design our plan, select reasonable fixed points by Genetic Algorithm, and use interpolation or fitting method to finish our job at last.

Key Words: Statistical Inference, Calibration, Three times spline interpolation method, Genetic Algorithm, Polynomial Fitting

1 引言

在工业生产中，经常需要对元件的特性进行标定，即对一系列的测试数据使用科学的方法进行分析，找出足以反应其特性的规律。但是往往由于一种元器件个体材料特性有所差异，若对每一个元件进行精准的定标显然要花费大量的时间和人力，不符合工业化生产的要求，因此寻求一种可行的简化标定过程的方法，降低时间和资金成本便十分重要。为研究这种过程的实现，我们应用统计推断的思想方法，使用启发式搜索算法，寻求优化标定的最佳方案。

1.1 课题内容与目标

某型投入批量试生产的电子产品，其内部有一个模块，功能是监测某项与外部环境有关的物理量（可能是温度、压力、光强等）。该监测模块中传感器部件的输入输出特性呈明显的非线性。本课题的主要内容是为该模块的批量生产设计一种成本合理的传感特性校准（定标工序）方案。

本课题的目标是：找到一个合适的定标方案，要求其经过给定的定标准确度评价函数在所有可行的方案中评价最优。

1.2 课题研究主要步骤

(1) 初步分析样本数据；
(2) 确定一种插值或拟合的方法；
(3) 使用搜索算法，确定合适的测量点个数及位置，找到一个高定标准确度的定标方案；
(4) 分析结果，优化或更换插值或拟合的方法，优化或更换搜索算法；

(5) 重复步骤(2)(3)(4)，直到得到满意的定标方案为止。

2 样本数据分析

2.1 建立数学模型

为了对本课题展开有效讨论，我们需要建立一个数学模型，对问题的某些方面进行必要的描述和限定。

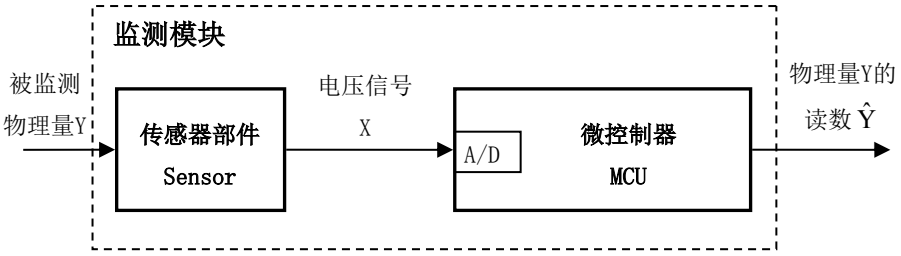

图 2-1 监测模块组成框图

监测模块的组成框图如图 2-1。其中，传感器部件的特性是我们关注的重点。传感器部件监测的对象物理量以符号 Y 表示；传感部件的输出电压信号用符号 X 表示。所谓传感特性校准，就是针对某一特定传感部件个体，通过有限次测定，估计其 Y 值与 X 值间一一对应的特性关系的过程。数学上可认为是确定适用于该个体的估测函数 $\hat{y} = f(x)$ 的过程，其中 x 是 X 的取值，$\hat{y}$ 是对应 Y 的估测值。

2.2 样本数据分析

在本课题的标准样本数据库中，共有 400 件样品，每个样品提供了 51 个测量数据点，测量点的输入值 X 的范围在 5 到 10 之间，每间隔 0.1 提供一个测量点，输出值即取值的范围不确定。通过对不同样品的函数曲线对比观察，可以总结出其函数关系具有以下几个特点：（1）单调递增；（2）在多数样品中函数表现为光滑曲线；（3）曲线具有非线性或者局部非线性；（4）不同样品之间存在差异。

3 评价标准

为评估和比较不同的定标校准方案，制定以下成本计算规则。

3.1 单点定标误差成本

$$s_{i,j} = \begin{cases} 0 & \text{if } |\hat{y}_{i,j} - y_{i,j}| \le 0.4 \\ 0.1 & \text{if } 0.4 < |\hat{y}_{i,j} - y_{i,j}| \le 0.6 \\ 0.7 & \text{if } 0.6 < |\hat{y}_{i,j} - y_{i,j}| \le 0.8 \\ 0.9 & \text{if } 0.8 < |\hat{y}_{i,j} - y_{i,j}| \le 1 \\ 1.5 & \text{if } 1 < |\hat{y}_{i,j} - y_{i,j}| \le 2 \\ 6 & \text{if } 2 < |\hat{y}_{i,j} - y_{i,j}| \le 3 \\ 12 & \text{if } 3 < |\hat{y}_{i,j} - y_{i,j}| \le 5 \\ 25 & \text{if } |\hat{y}_{i,j} - y_{i,j}| > 5 \end{cases} \quad (3\text{-}1)$$

单点定标误差的成本如式（3-1），其中 $y_{i,j}$ 表示第 i 个样本之第 j 点 Y 的实测值，$\hat{y}_{i,j}$ 表示定标后得到的估测值（读数），该点的相应误差成本以符号 $s_{i,j}$ 记。当估测值与实测值的

偏差较大时，相应的误差成本较大，这是对过大偏差的惩罚性。

3.2 单点测定成本

实施一次单点测定的成本以符号 q 记。本课题指定 q=12。

3.3 对单一样品的定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \tag{3-2}$$

对样本 i 总的定标成本按式（3-2）计算，式中 n_i 表示对该样本个体定标过程中的单点测定次数。

3.4 校准方案总体成本

按式（3-3）计算评估校准方案的总成本，即使用该校准方案对标准样本库中每个样本个体逐一定标，取所有样本个体的定标成本的统计平均。

$$C = \frac{1}{M} \sum_{i=1}^{M} S_i \tag{3-3}$$

总成本较低的校准方案，认定为较优方案。要值得注意的是，总成本最低的方案，对特定的样品而言不一定是最低成本。

4 拟合方法的讨论

4.1 三次样条插值法

4.1.1 方法介绍

插值法的基本思想是构造一个简单函数 y=G(x)作为 f(x)的近似表达式，以 G(x)的值作为函数 f(x)的近似值，并且要求 G(x)在给定数据点 x_i 的值与 f(x)的值相同，即 $G(x_i)=f(x_i)$，通常称 G(x)为 f(x)的插值函数，x_i 为插值节点。

4.1.2 样本数据拟合

$$y=spline(x_0,y_0,x) \tag{4-1}$$

我们使用 MATLAB 中内嵌函数（式 4-1）对样本数据库中的某个数据点进行三次插值拟合，所得拟合曲线如图 4-1。

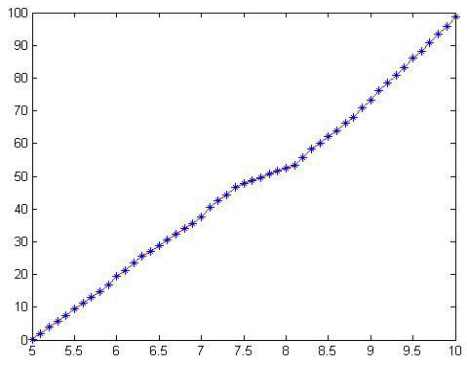

图 4-1 三次样条插值拟合曲线

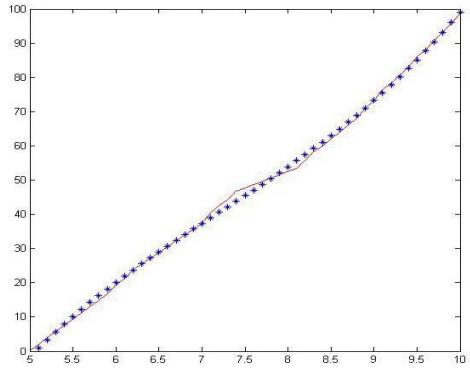

图 4-2 三次多项式拟合曲线

4.2 三次多项式拟合法

4.2.1 方法介绍

多项式拟合是一种最基本的拟合方式，对于给定数据(x_i，y_i)，在给定的函数簇Φ中，求属于Φ的 f(x)，使误差 $r_i=f(x_i)-y_i$ 最小，这样所求的函数 f(x)即为拟合出来的多项式函数。使用三次多项式拟合，其$\Phi(x)=ax^3+bx^2+cx+d$。

4.2.2 样本数据拟合

我们使用 MATLAB 中内嵌函数（式 4-2）对样本数据库中的某个数据点进行三次多项式拟合，所得拟合曲线如图 4-2：

$$y=polyfit(x_0,y_0,m) \tag{4-2}$$

用 MA 其中 x_0, y_0 为拟合点坐标，m 为多项式拟合的次数，这里 m=3。

5 遗传算法

5.1 遗传算法简介

遗传算法（Genetic Algorithm）是一类借鉴生物界进化规律（如适者生存、优胜劣汰遗传机制）演化而来的随机化仿生搜索方法。其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定其规则。

5.2 遗传算法一般算法

1. 建立初始状态：初始种群从样本库中随机选出 N 个个体，成为第一代 P(0)。此时设置进化代数计数器 t=0，设置最大演化代数 T。

2. 个体评价：计算群体 P(t)中各个个体的适应度。

3. 繁殖运算：通过选择算子，将得到优化的个体直接遗传到下一代或者通过交叉配对产生新的一代。这个操作是建立在个体评价的基础上的。

4. 交叉运算：把两个父代个体的部分结构相互替换重组而生成新的子代个体。

5. 变异运算：将变异算子应用于群体，即对群体中个体的基因值进行一定概率的改动，之后群体通过选择、交叉运算后得到下一代。变异算法有：实值变异、二进制变异。其算子的操作步骤为：(1) 对群体中所有个体以事先设定的变异概率来决定是否进行变异； (2)对变异的个体随机选择变异位点进行变异。

6. 终止条件判断：当 t=T 时，则以进化过程中所得到拥有最大适应度的个体作为最优解输出，并终止运算。

遗传算法的基本流程可以由图 5-1 所示流程图来表示。

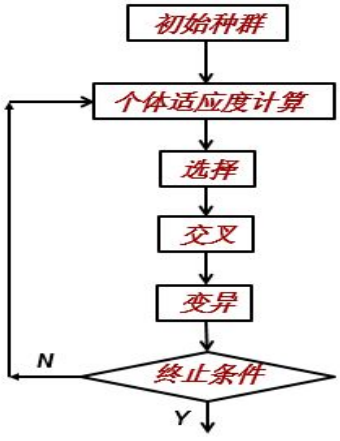

图 5-1 遗传算法流程图

6 遗传算法的实现

这里罗列一些关键算法的实现，具体实现过程的代码见附录。本次试验在每一个样本中取 7 个点作为一个生物个体，并且计算其适应度。其中一些参变量的设置如下：

popSize：代表种群大小；
gene：代表种群；
cost：代表适应度；
generation：代表进化代数，设置为 100；
crossRate：代表交叉概率，设置为 0.85；
mutationRate：代表变异概率，设置为 0.01；

以下为算法的实现过程：（按照图 5-1 所示流程进行，即主函数的实现）

1. 读入所提供的样本数据。

2. 用随机函数随机选取 7 个点，得到一个数组，对这个数组中的元素进行升序排列，将所得到的有序数组作为个体基因的染色体。

3. 初始化种群函数 gene_init(popSize)随机生成 popSize 个生物个体的染色体，存在种群列表 gene 中作为初始种群。

4. 个体适应度函数 adapt(gene,y,popSize)计算个体的平均成本作为其适应度。函数中内嵌函数 errorcost(dy)使用式 3-1 所规定的的误差计算方案计算单个个体误差成本。将所计算的适应度存放在适应度列表 cost 中。

5. 自然选择函数 select(gene,cost,popSize)根据适应度的要求选择能繁殖后代的个体，储存进 gene 中作为下一代的父代。因为自然选择所保留的生物个体是在自然中适应度较高的那部分，因此设置该个体 i 被选择的概率为：

$$p_i = \frac{\text{第}i\text{个个体的适应度}}{\text{总适应度}}$$

将每个个体被选择的概率存在一个列表 s 中，然后使用二分法查找函数找到适应度高的个体存在新的 gene 中，作为新的种群，即为下一代的父代。

6. 交叉过程函数 generate(gene,popSize,crossRate)。在选择出种群父代后，在其内进行交叉。我们采用单点交叉的方式，在基因段中随机一处断裂并与另一基因进行重组，设定交叉概率 crossRate=0.85。将父代随机排列并按照 i 和(popSize-i+2)进行配对，其中 popSize=100。若二者进行交叉，则随机产生一个断裂点，将二者基因重组后加入新的种群；若二者不进行交叉，则将二者输入新的种群。

7. 变异过程函数 mutate(gene,popSize,mutationRate)即将基因中的碱基进行随机替换。由于变异概率较小，且仅仅提供基因的多样性，所以我们设定变异的概率 mutationRate=0.01。由于种群中每一个个体都可能发生突变，且种群是由二维矩阵表示的，因此变异过程需要使用二重循环来实现。

8. 主函数按照遗传算法的基本流程，先调用样本数据，然后使用初始化种群函数 gene_init(popSize)生成初始种群，然后进入第一代循环；在每一代循环中，先调用个体适应度函数对种群个体的适应度进行计算，然后调用自然选择函数选择出适应度较高的个体作为下一代个体的父代，并进行交叉和变异，之后判断是否达到预设的代数，若未达到，则再次进入相同的循环，这次对象即为上次循环所生成的父代。每次循环都会调用 display 函数来输出结果，便于观察。

7 数据分析

本课题要求选取最少数目的点以最小的代价拟合出曲线，若所选取的点数太大会引起实验成本的增加，若选取点数太小则无法准确拟合曲线。在开始时随机选取实验数据点，然后计算所选点的适应度，删除适应度较低的点，经过 100 代的循环，最终选取的点数为 6 到 7 个。 我们选择 50 到 100 不同的种群大小，分别运行程序，发现随着种群大小的增加，达到稳定状态即最优解所需要的代数越来越少，而在种群较小时，可能存在不稳定的状况导致较大的误差，因此我们选择种群大小为 100 的种群。通过对比三次多项式拟合与三次样条插值拟合的结果，发现后者所得结果更优，因此选择后者作为拟合方式。由于遗传算法下所得的最优解为一个解空间且答案并不唯一，所以我们通过多次运行程序寻找其中所出现的尽可能小的解。经过 100 次运行，挑选其中较小的 10 个结果，如表 7-1 所示。

编号	结果	成本
1	1 9 19 27 34 44 51	96.604250
2	1 8 20 27 35 44 51	96.691250
3	1 8 20 26 34 44 51	96.715500
4	1 8 19 26 33 44 51	96.809250
5	1 9 19 26 35 44 51	96.816750
6	1 9 19 27 35 45 51	96.898250
7	1 9 20 26 34 43 51	96.905000
8	1 9 19 26 33 44 51	96.708750
9	1 8 19 27 34 44 51	96.725250
10	1 9 18 26 34 44 51	96.916250

表 7-1 程序运行结果表

其中编号 1 选取最优解特征点的过程如表 7-2 所示，在 67 代后找到最优解，最终选取的点数为 7 个。

代数	特征点
1	1 5 10 12 13 22 29 31 41 51
3	1 5 12 13 22 29 31 41 51
4	1 5 10 13 22 29 31 41 51
5	1 5 12 22 29 31 41 51
7	1 5 12 22 29 41 51
9	1 5 22 29 41 51
11	1 5 16 22 29 41 51
15	1 5 16 23 29 38 44 51
18	1 5 16 23 29 34 44 51
21	1 5 16 29 34 44 51
24	1 10 16 29 34 44 51
25	1 9 16 29 34 44 51
52	1 9 19 29 34 44 51
67	1 9 19 27 34 44 51
100	1 9 19 27 34 44 51

表 7-2 选取特征点过程表

8 结论

经过以上分析，我们可以通过遗传算法得到本试验的最优解。
最终我们得到 7 个特征点为：［1　9 19 27 34 44 51］
其所对应的最小成本为：96.60425

9 拓展部分

在拓展部分中，我们了解了另一种启发式搜索算法——模拟退火算法。

模拟退火算法的基本步骤如图 8-1 所示：

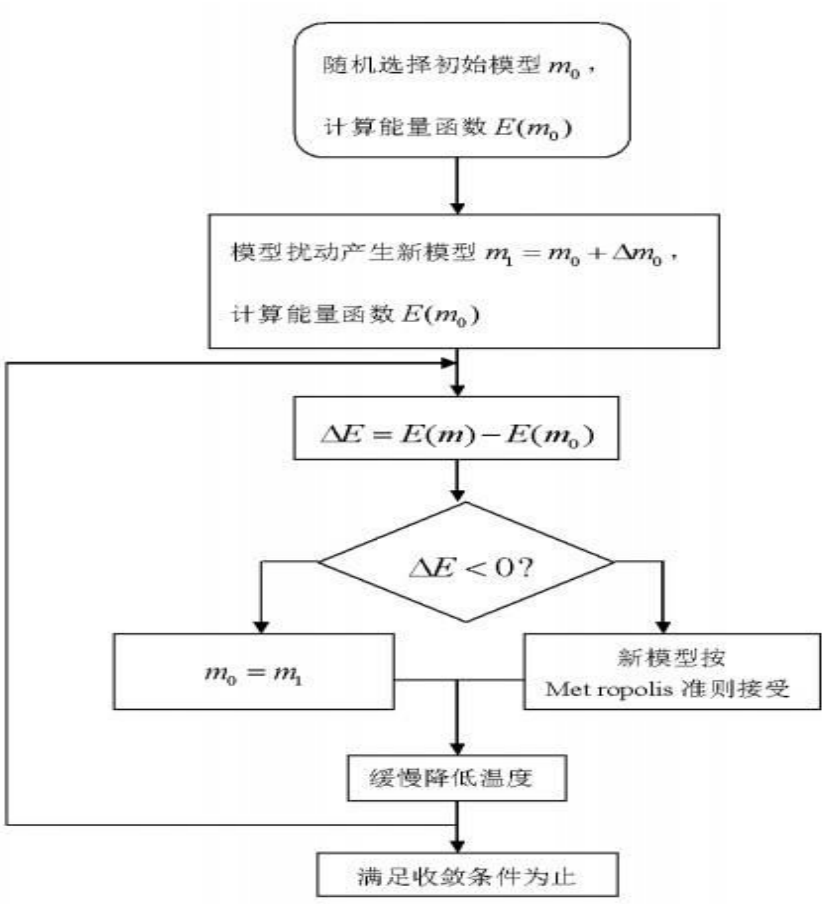

图 8-1 模拟退火算法流程图

模拟退火算法的基本思想是：

粒子在温度 T 时趋于平衡态的概率为 e-ΔE/(kT)，其中 E 为温度 T 时的内能，ΔE 为其改变量,k 为玻尔兹曼常数。用固体退火模拟组合优化问题，将内能 E 模拟为目标函数值 f,温度 T 演化成控制参数 t,即得到解组合优化问题的模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解。

从模拟退火算法的基本思想和流程来看，我们不难观察到，退火算法先是随机选择一个初始样本模型，对之计算能量函数，相当于在遗传算法中所计算的适应度。然后对这个样本进行扰动，得到新的样本模型，计算此模型的能量，然后与初始样本进行比较，若温度降低了，即能量减少，相当于遗传算法中适应度较高的个体，则接受这个扰动差生的新样本模型，如果能量未减少，则用 Metropolis 准则来判断能否接受这个新的样本模型，经过一次过程，

温度下降，判断是否降低到初始设置的目标温度，若达到则结束循环得到最优解，若未达到则进入下一次循环，继续退火。

可以看到，退火算法是以单一个体为目标进行优化的，每一次优化都需要与设置的优化条件比较，比较繁琐，时间耗费较多，但所得最优解比较精确。遗传算法是通过对种群中多个个体同时进行遗传交叉和变异以达到优化的目的，最终得到一个近似的解空间，可以从中选取最优解，比较节省时间。二者各有长处，模拟退火算法更易于理解。

10 参考文献

[1] 上海交通大学电子工程系. 统计推断讲座 1,2,3. ftp://202.120.39.248.

[2] 上海交通大学电子工程系. 课程资料. matlab_tutorial. PDF.

[3] 百度百科. http://baike.baidu.com

11 致谢

感谢李老师和袁老师以及助教老师的悉心指导，为我们在课程中的学习和探索带来了巨大的帮助，也为我们的算法和论文的书写提供了宝贵的修改意见，帮助我们更好的理解遗传算法的本质以及其在统计推断中的应用。同时，也感谢学院所开设的这门课程，让我们在实践中学习到了许多知识，掌握了新的能力。

12 参考他人报告或代码的申明

统计推断课程，2015 年秋季学期第 26 组，成员：王天祎 学号 5140309214，李琰 学号 5140309225，在报告编写过程中，以下方面参考了往届报告，现列表说明：

主要参考项目	说明
代码方面	《统计推断在数模模数转换中的应用》，于哲昆，2014 年秋季学期，组号 43 在该组报告附录提供的程序代码基础上，进行了部分修改。
算法描述方面	《统计推断在数模转换系统中的应用》，赵忆漠，2014 年秋季学期，组号 28 参考了该组报告的算法描述文字。
拓展问题研究	《统计推断在数模转换系统中的应用》，宋凯敏，2014 年秋季学期，组号 60 参考了该组对模拟退火算法基本思想以及流程图。

除了以上注明的参考内容，和报告中列出的引用文献注解，本报告其他部分都不包含任何其他个人或集体已经发表或撰写过的作品成果。

13 附录

附录 1： 遗传算法主函数

```
data=csvread('20150915dataform.csv');       %读入原始数据

%设置数据
popSize=100;                                %种群大小
crossRate=0.85;                             %交叉概率
mutationRate=0.01;                          %突变概率
generation=100;                             %进化代数

y=zero(400,51);
y(1:400,:)=data(2:2:800,:);
gene=gene_init(popSize);                    %随机产生初始种群
for g=1:generation
    display(g);                             %将每一代的基因显示出来
    cost=adapt(gene,y,popSize);             %计算这一代个体的成本
    gene=select(gene,cost,popSize);
    gene=generate(gene,popSize,crossRate);
    gene=mutate(gene,popSize,mutationRate);
    display(find(gene(1,:)==1));
end
xx=find(gene(1,:)==1);
assess(xx,y);                               %计算平均成本
```

附录 2： 遗传算法主要函数

（1）随机产生初始种群

```
                                            %初始化种群
function out = gene_init(popSize)

out = round(rand(pop,51)-0.2);
out(:,1) = 1;
out(:,51) = 1;

end;
```

（2）二分查找

```
function [out]=search(in,s,l,r)

mid=floor((l+r)/2);
if in<=s(mid)
    if in>s(mid-1)
        out=mid-1;
    else
        out=search(in,s,l,mid);
```

```
    end
else
    if in<=s(mid+1)
        out=mid;
    else
        out=search(in,s,mid,r);
    end
end
end
```

（3）交叉保留

```
function out=generate(gene,popSize,crossRate)

for i=2:floor(popSize/2+1)
    out=gene;
    mid=floor(rand()*50)+1;
    p=rand();
    if p<=crossRate
        out(i,1:mid)=gene(popSize-i+2,1:mid);
        out(popSize-i+2,1:mid)=gene(i,1:mid);
        out(i,mid+1:51)=gene(popSize-i+2,mid+1:51);
        out(popSize-i+2,mid+1:51)=gene(i,mid+1:51);
    end
end
end
```

（4）自然选择

```
function out = select(gene,cost,popSize)        %保留适应度较高的成员，作为父代

out=zeros(popSize,51);
dcost=max(cost)-cost;
s0=sum(costd);
s=zeros(popSize+1);
s(1)=0;
s(popSize+1)=1;
s(2:popSzie)=sum(dcost(1:popSzie-1))/s0;
for i=2:popSzie
    p=rand();
    j=search(p,s,1,popSzie+1);
    out(i,:)=gene(j,:);
end
sort0=[[1:popSize]',cost];
sort0=sortrows(sort0,2);
out(1,:)=gene(sort0(1,1),:);
```

```
end
```

（5）基因突变

```
function out = mutate(gene,popSize,mutationRate)  %使用双层循环
out=gene;
for i=2:popSize;                                  %从第二代到第 99 代外循环
    for j=2:50;
        p=rand();
        if p<=pm
            out(i,j)=~out(i,j);
        end
    end
end

end
```

附录 3： 成本计算函数

(1) 计算每个个体的平均成本

```
function adapt(gene,y,popSize)
out=zeros(popSize,1);
x=5:0.1:10;
for i=1:popSize
    c=sum(gene(i,:)==1);              %测试点数量
    pos=find(gene(i,:)==1);           %测试点位置
    xx=5+(pos-1)*0.1;                 %测试点 x 值
    yy=y(:,pos);                      %测试点 y 值
    f=spline(xx,yy);                  %插值拟合
    dy=ppval(f,x)-y;                  %理论值与实际值的差
    out(i)=12*c+errorcost(dy)/400;    %单个个体平均成本
end
min(out)
mean(out)
end
```

(2) 计算单个个体误差成本　　%使用 3.1 中设定的评价标准

```
function [out] = errorcost(dy)

t=abs(dy);

t0=sum(sum(t<=0.4));
t1=sum(sum(t<=0.6))-t0;
t2=sum(sum(t<=0.8))-t0-t1;
t3=sum(sum(t<=1))-t0-t1-t2;
```

```
t4=sum(sum(t<=2))-t0-t1-t2-t3;
t5=sum(sum(t<=3))-t0-t1-t2-t3-t4;
t6=sum(sum(t<=5))-t0-t1-t2-t3-t4-t5;
t7=sum(sum(t>5));
out=0.1*t1+0.7*t2+0.9*t3+1.5*t4+6*t5+12*t6+25*t7;

end
```

(3) 计算平均成本

```
function [out] = assess(xx,y)

out=length(in)*12;
x=5:0.1:10;
xx=5+(in-1)*0.1;
yy=y(:,in);
f=spline(xx,yy);
dy=ppval(f,x)-y;
out=out+errorcost(dy)/400;
s=sum(dy.^2);

fid=fopen('ans.txt','a');
fprintf(fid,'position: [ ');
fprintf(fid,'%2d ',in);
fprintf(fid,'] mean_cost: %7f\n\n',out,s);
fclose(fid);

end
```