

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 19 组，成员陈昶安，学号 5140309179，田丰，学号 5140309174，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》，艾玥祺，2014 年秋季学期，组号 20 在该组报告附录提供的程序代码基础上，进行了少量修改。

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。

统计推断在数模模数转化中的应用

组号：19 姓名：陈昶安 学号：5140309179 姓名：田丰 学号：5140309174

摘要： 本组使用遗传算法从大量数据中选取合适测试点，并用三次多项式拟合与三次样条插值法对测试点进行分析，为某型电子产品内部的一个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。

关键词： 三次多项式拟合，三次样条插值，遗传算法，拟合

1. 词汇与符号约定

1.1 词汇表一

名称	含义
拟合	科学或工程上可以通过实验等方法获得关于某个问题的若干离散数据，依靠数学方法，使用连续函数（也就是曲线）或者更加密集的离散方程尽量逼近（即最小二乘意义上的差别最小化）这些已知离散数据点集，此过程称为拟合。
插值（内插）	通常有两种含义。一是指曲线必须通过若干已知离散数据点的一种拟合；二是指利用拟合得到的函数曲线在某区间内的值，作为原本数值未知的点的近似取值。本课程中，“插值”一词主要指上述第二种含义。
搜索算法	利用计算机的高性能运算能力，有目的地检验一个问题之解空间包含的部分或全部可能情况，从而求得问题的优化解的做法。
启发式搜索	在搜索问题解空间时，对当前已搜索的位置进行评估，寻找认为最好的下一步搜索方向，从这个（或这些）方向进行搜索直到目标。启发式搜索可以避免简单穷举式搜索效率低（甚至不可完成）的弊端。
三次样条插值	又称 Spline 插值，是通过一系列形值点的一条光滑曲线，数学上通过求解三弯矩方程组得出曲线函数组的过程。
遗传算法	是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

1.2 词汇表二（课程自定义词汇）

名称	含义
单点测定	在本课题中，对 X 取某一定值时 Y 的真

	实取值进行测量确定的过程。
单点测定成本	完成单点测定要付出一定成本，称为单点测定成本。
单点定标误差成本	在本课程中，系统完成定标后，对物理量 Y 的单点读数（即系统对 Y 的估测值）与该点 Y 实测值之间存在误差，按一定规则把该误差折算为一种成本。

1.3 符号约定

符号	含义
X	传感部件的输出电压
Y	传感部件的输入，即被监测物理量
\hat{Y}	监测模块的输出，即监测模块将传感部件输出 X 所转换成的读数（对 Y 的估测值）
$y_{i,j}$	第 i 个样本之第 j 点对应的 Y 实测值（来自标准样本数据库）
$\hat{y}_{i,j}$	第 i 个样本之第 j 点对应的 Y 估测值
$s_{i,j}$	第 i 个样本之第 j 点对应的单点定标误差成本
q	单点测定成本
n_i	对样本 i 定标过程中的单点测定次数
S_i	对样本 i 的定标成本
M	标准样本数据库中的样本总数
C	基于标准样本数据库评价一个校准方案，算得的该方案总体成本

2. 课题重述

本课题欲对某型电子产品内部的一个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，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。本组通过对 400 个样品数据 51 个观测点进行处理，用遗传算法来选取合适的测试点，然后选取三次多项式拟合与三次样条插值方法进行拟合，并比较两种方法得到的输入输出特性下的成本大小，从而得出较优方案。

3. 建模与解模

3.1 建模

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

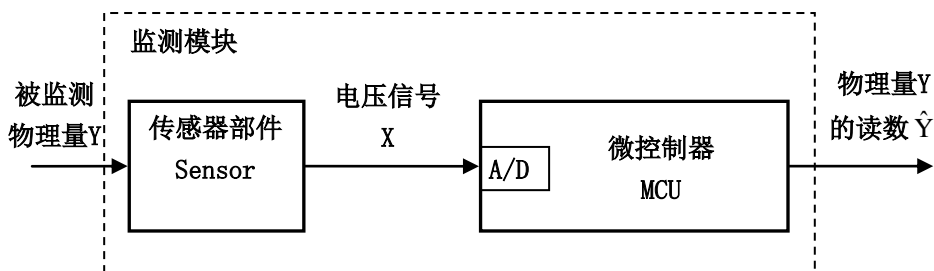


图 1 监测模块组成框图

监测模块的组成框图如图 1。其中，传感器部件（包含传感器元件及必要的放大电路、调理电路等）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，该电压经模数转换器（ADC）成为数字编码，并能被微处理器程序所读取和处理，获得信号 \hat{Y} 作为 Y 的读数（监测模块对 Y 的估测值）。

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。

考虑实际工程中该监测模块的应用需求，同时为便于在本课题中开展讨论，我们将问题限于 X 为离散取值的情况，规定

$$X \in \{x_1, x_2, x_3, \dots, x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, \dots, 9.9, 10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i = f(x_i)$ ， Y 实测值记为 y_i ， $i = 1, 2, 3, \dots, 50, 51$ 。

3.1.1 传感部件特性

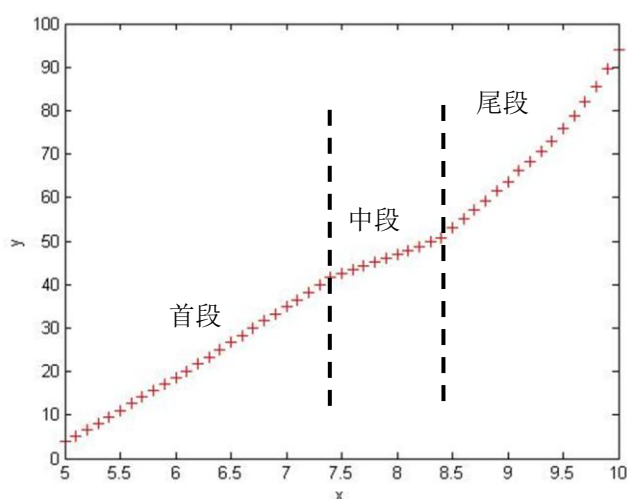


图 2 传感特性图示

一个传感部件个体的输入输出特性大致如图 2 所示，有以下主要特征：

- Y 取值随 X 取值的增大而单调递增；

- X 取值在[5.0, 10.0]区间内，Y 取值在[0, 100]区间内；
- 不同个体的特性曲线形态相似但两两相异；
- 特性曲线按斜率变化大致可以分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段；
- 首段、中段、尾段单独都不是完全线性的，且不同个体的弯曲形态有随机性差异；
- 不同个体的中段起点位置、终点位置有随机性差异。

为进一步说明情况，图 3 对比展示了四个不同样品个体的特性曲线图示。

3.1.2 标准样本数据库

前期已经通过试验性小批量生产，制造了一批传感部件样品，并通过实验测定了每个样品的特性数值。这可以作为本课题的统计学研究样本。数据被绘制成表格，称为本课题的“标准样本数据库”。

该表格以 CSV 格式制作为电子文件。表格中奇数行存放的取值，偶数行存放对应的取值。第 $2i - 1$ 行存放第 i 个样本的 X 数值，第 $2i$ 行相应列存放对应的实测 Y 数值。

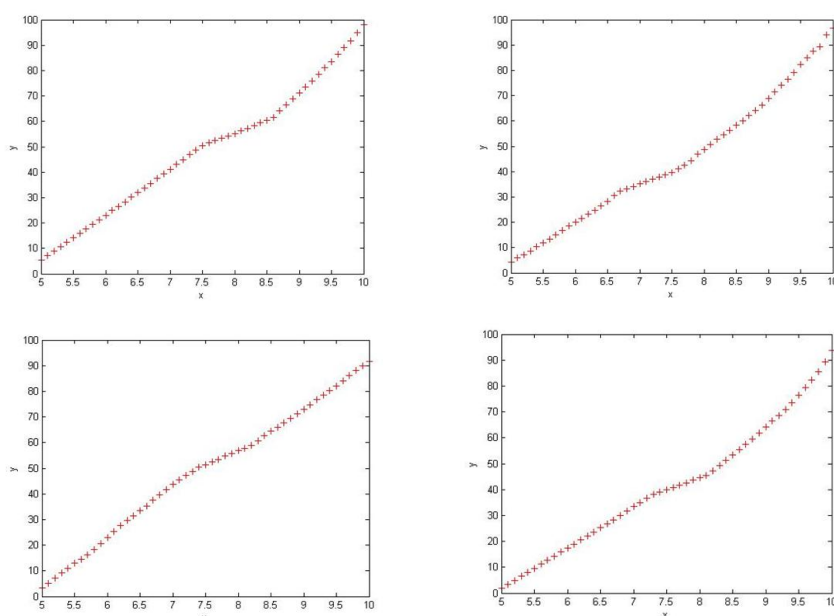


图 3 四个不同样本个体特性图示对比

3.1.3 成本计算

为评估和比较不同的校准方案，特制定以下成本计算规则。

- 单点定标误差成本

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

单点定标误差的成本按式(1)计算, 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。

- 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 $q=12$ 。
- 某一样本个体的定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (2)$$

对样本 i 总的定标成本按式(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。

- 校准方案总成本

按式(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。

$$C = \frac{1}{M} \sum_{i=1}^M S_i \quad (3)$$

总成本较低的校准方案, 认定为较优方案。

3.2 解模

3.2.1 估测成本大致区间

为了方便程序的编译, 我们首先随机取估定点, 并应用遗传算法, 三次样条插值与三次多项式拟合计算出估测成本

3.2.1.1 利用随机区间来估定点

在测试阶段, 我们随机选取了 6 个区间, 用 for 循环暴力求解。

3.2.1.2 利用遗传算法求测试点

在本例中, 我们在选择测定点的组合方案时, 选取的启发式搜索算法为遗传算法。(注: 此处都采取 6 点取样。)

遗传算法:

遗传操作是模拟生物基因遗传的做法。在遗传算法中，通过编码组成初始群体后，遗传操作的任务就是对群体的个体按照它们对环境适应度(适应度评估)施加一定的操作，从而实现优胜劣汰的进化过程。从优化搜索的角度而言，遗传操作可使问题的解一代代优化，并得出最优解。

遗传算法由编解码、个体适应度评估和遗传运算三大模块构成，而遗传运算又包括染色体复制、交叉、变异甚至倒位等。在遗传算法中，定义种群或群体为所有编码后的染色体集合，表征每个个体的是其相应的染色体。

i. 编码

遗传算法的编码有浮点编码和二进制编码两种，这里只介绍二进制编码规则。二进制编码既符合计算机处理信息的原理，也方便了对染色体进行遗传、变异和突变等操作。

ii. 解码

解码的目的是为了将不直观的二进制数据串还原成二进制。遗传算法的编码和解码在宏观上可以对应生物的基因型和表现型，在微观上可以对应 DNA 的转录和翻译两个过程。

iii. 交叉

用随机数产生一个或多个交叉点位置，然后两个个体在交叉点位置互换部分基因码，形成两个子个体。

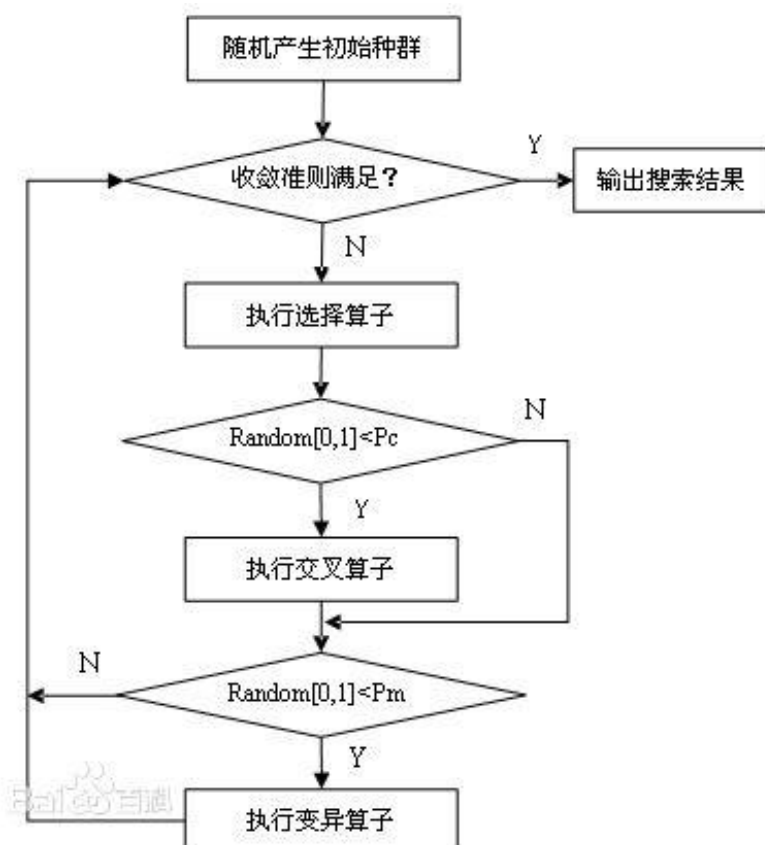
iv. 突变

为了避免在算法迭代后期出现种群过早收敛，对于二进制的基因码组成的个体种群，实行基因码的小几率翻转，对于二进制编码即 0 变为 1，而 1 变为 0。

v. 个体适应度

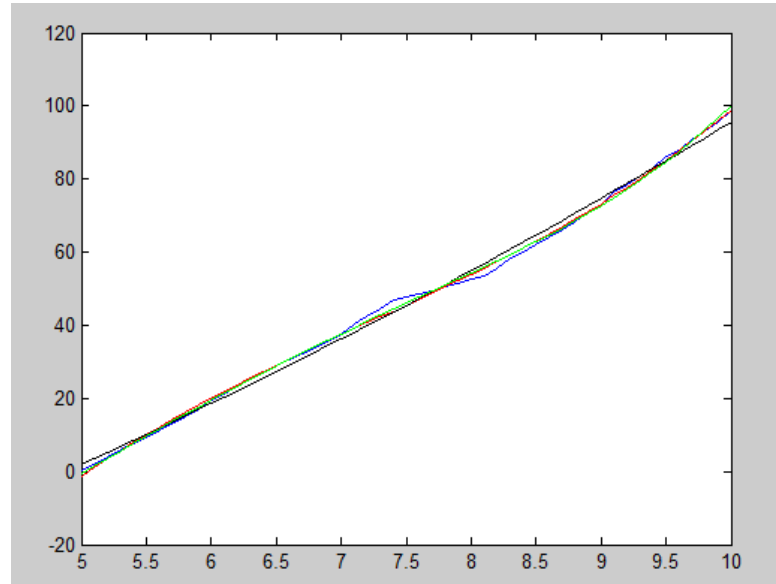
将一个区域分为若干块，落在各个区域的几率对应遗传算法中各条染色体被遗传到下一代的概率。遗传算法依照与个体适应度成正比的几率决定当前种群中各个个体遗传到下一代群体中的机会。个体适应度更大的个体更容易被遗传到下一代。

编码的基本框架：



3.2.1.3 根据测试点进行拟合

<方法一>三次多项式拟合（polyfit 函数）【详细代码见附录一】



注：我们组对多项式次数进行了估测，如图（蓝色为原曲线，黑红绿依次为 2 次、3 次、4 次多项式拟合情况，可以看出，3 次拟合较为符合原曲线。

<方法二>三次样条插值（spline 函数）【详细代码见附录二】

3.2.1.4 尝试求解

在随机取点的情况下，利用三次样条插值拟合，得出的结果为：

最佳取点	最小成本
[3 12 22 31 43 50]	94.8983

3.2.2 正式测定成本（在前面估测完最小成本的大致数量后，开始正式测定成本）【详细 matlab 代码见附录三】

3.2.2.1 取定测试点

在初稿的随机采样取点时，我组尝试了 5 点取样法，但是发现误差异常地大，最终最低成本在 100 左右，远远超过预期最低成本。所以决定采用取 6 个点或 7 个点的方法，在完善的程序过程中，发现当取 6 个点时比较适合，所以后期我们主要是修复和完善取 6 个点时的程序，通过改变变异等的概率来达到缩小成本的效果。

3.2.2.2 利用遗传算法计算最小成本

I. 程序基本变量

变量	含义
point	取点数
size	每一代的种群数
gender	当前操作的代数
zong	要循环的总代数

II. 程序思想

a. 先从 1-51 中随机取 6 个点，重复 200 组（即有 200 个种群），再由 create 函数生成二维点列 xx。进入代数循环后，将 1-51 范围内的点一一对应地转换到 5.0-10.0 的范围内，

将新得到的二维点列命名为 x (x 的第一个参数 $size$ 代表组数, 第二个参数 $point$ 代表该组的第几个数)。因为每一个 x 对应着 400 个值, 所以需构建一个三维点列 y 。然后就是通过前面的讨论用 $spline$ 三次样条插值法来拟合出曲线 (这里因为 y 是三维点列, $y(k, j, :)$ 不能代表一维的, 所以利用 $ytmp$ 储存这些值, 变为一维的), 并且求出在 x 点处拟合函数的值, 这就是第一代的拟合结果。

b. 在得到了第一代结果的前提下, 首先要找出前一代 (这里是第一代) 的最佳取点组 $mini$ (成本最低的一组点)。接着利用公式 (存活率 = $10000 / (\text{成本})^3$) 计算存活率, 这样能使成本低组存活率较高, 用 $live$ 函数实现。得到存活率后, 将其分配到 0-1 区间内, 并且通过 $rand$ 随机值来根据概率生成下一代的未交叉变异的初始种群, 这样的做法会使最终的结果稳定在几个种群 (选点方法) 不会有较大偏差。

c. 接着是交叉和变异。首先将每一组的选点由十进制转换成二进制, 用三维数组 $bincode$ 保存, 并将上一代的 200 组数据重新按二进制的方式存储在 $select$ 里面。使用 $crosssample$ 函数对 $select$ 进行交叉, 在这里并没有设置几率, 而是用了 $for k=1:2: size$ 来做, 相当于有一半的几率交叉。交叉过程中会产生大于 51 的点, 这里先不予以处理, 待变异发生后再处理畸形就行了。接着用 $trans$ 函数实现变异过程, 就是通过一定几率 (在这里我组尝试了 2% 和 1%, 发现结果都异常的地稳定) 选择 1-51 中的一个点, 转换为二进制, 从二进制的六位中随机选择一个点, 改变该点的值 (即取非)。

d. 最后将这些新生成的二进制片段重新转换成十进制, 对于转换后超过 51 的点则重新随机出一个 1-51 范围内的点代替。然后就是去重 (否则易造成 $spline$ 拟合的 bug, 因为生成的是函数, 不能一个 x 对应多个 y , 所以必须保证 x 的一维向量中没有重复)。接着就是像第一代那样按照升序排列好, 这就是下一代的初始种群, 接着就是下一代循环。

3.2.2.3 最终结果

利用遗传算法运行程序 200 代后, 我们得到了最终的结果为:

最佳取点	最小成本
[2 10 21 30 41 49]	93.4060

4. 参考文献

[1] Matlab 读取 CSV 文件:

http://blog.sina.com.cn/s/blog_a0246c110101vlk3.html

[2] 遗传算法及 Matlab 实现

http://wenku.baidu.com/link?url=2EF9r0eVFafDnJmuu31qbDF61rVDq00fI1jImoy5klpm63X8lejWNvTSMP0qrmXsmeFew5cpt7ZOewA_VQ2bbq3qP0zwrBrnl4b8dXzbe

[3] 百度百科 遗传算法

<http://baike.baidu.com/link?url=RBI5gvTMVREkQ38WcPHwtXggkJeq-1jbsS9QWTPGEnFWzCAfvmBh25wB0BVI2KafbZfMuJpP3s3L6BvYk3Dq-a>

[4] 《插值与拟合》pdf 基本数值计算方法课程 上海交大 顾卫华

[5] 统计推断 PPT

[6] 《MATLAB 在数学建模中的应用》 遗传算法基本原理 卓金武

附录

注：以下代码均通过了 matlab 测试，为方便，工作文件我们组保存在 c:\1.csv 中
附录一（尝试求解）随机样条插值：

```
clear all;
close all
clc;
data=csvread('c:\1.csv',0,0,[0,0,799,50]);
sum=0;
min=10000;
for k1=1:3
    for k2=7:13
        for k3=17:24
            for k4=27:35
                for k5=37:43
                    for k6=45:50
                        key=[k1 k2 k3 k4 k5 k6];
                        sum=jisuan(data,key);
                        if sum<min
                            min=sum;;
                            disp(key);
                            disp(min)
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
```

jisuan 函数：

```
function sum = jisuan( data , key )
xi=zeros(1,51);
sum=0;
for o=1:400
    keyx=[data(2*o-1,key(1)) data(2*o-1,key(2)) data(2*o-1,key(3)) data(2*o-1,key(4))
data(2*o-1,key(5)) data(2*o-1,key(6)) ];
    keyy =[data(2*o,key(1)) data(2*o,key(2)) data(2*o,key(3)) data(2*o,key(4))
data(2*o,key(5)) data(2*o,key(6)) ];
    for m=1:51
        xi(m)=data(2*o-1,m);
    end;
    y = interp1(keyx,keyy,xi,'spline');
    cha=y-data(2*o,:);
    score=72;
```


jisuanpoly 函数:

```
function sum = jisuanpoly( data , key )
xi=zeros(1,51);
sum=0;
for o=1:400
    keyx=[data(2*o-1,key(1)) data(2*o-1,key(2)) data(2*o-1,key(3)) data(2*o-1,key(4))
data(2*o-1,key(5)) data(2*o-1,key(6)) data(2*o-1,key(7))];
    keyy =[data(2*o,key(1)) data(2*o,key(2)) data(2*o,key(3)) data(2*o,key(4))
data(2*o,key(5)) data(2*o,key(6)) data(2*o-1,key(7))];
    for m=1:51
        xi(m)=data(2*o-1,m);
    end;
    p3 =polyfit(keyx,keyy,3);
    cha = polyval(p3,data(2*o-1,:))-data(2*o,:);
    score=72;
    for t=1:51
        if abs(cha(t))<=0.4 score=score+0;
        elseif abs(cha(t))<=0.6 score=score+0.1;
        elseif abs(cha(t))<=0.8 score=score+0.7;
        elseif abs(cha(t))<=1 score=score+0.9;
        elseif abs(cha(t))<=2 score=score+1.5;
        elseif abs(cha(t))<=3 score=score+6;
        elseif abs(cha(t))<=5 score=score+12;
        else score=score+25;
        end;
    end;
    sum=sum+score;
end;
sum=sum/400;
end
```

附录三 核心算法——遗传算法

主程序

```
data=csvread('c:\1.csv');
point=6;
Zong=200;
Gender=1;
size=100;
xx=zeros(size,point);
for k=1:size
    z=create(1,51,point); %利用create函数生成size组point个1—51中的随机点
    for j=1:point
```

```

        xx(k, j)=z(j);
    end
end
while (Gender<=Zong)      %循环体 走代数
x=zeros(size, point);    %将随机选的点按题目要求转化到5.0—10.0范围内
for k=1:size
    for j=1:point
        x(k, j)=5+(xx(k, j)-1)*0.1;
    end
end
y=zeros(size, 400, point); %因为一个x对应400个y，所以y的维数比x多一维
for g=1:size
    for j=1:400
        for k=1:point
            y(g, j, k)=data(j, xx(g, k));
        end
    end
end
ys=zeros(size, 400, 51); %计算y的三次样条插值
for uu=1:51
    xxd(uu)=5+(uu-1)*0.1;
end
for k=1:size
    for j=1:400
        xtmp=x(k, :);
        tmp=y(k, j, :); %y(k, j, :)并不是简单的行向量，所以要另开一个tmp保存成行向量
        for u=1:point ytmp(u)=tmp(u); end
        pp(j)=spline(xtmp, ytmp);
        kt=ppval(pp(j), xxd);
        for g=1:51
            ys(k, j, g)=kt(g);
        end
    end
end
jieguo=result(size, ys, data, point); %利用题目信息计算总成本
mini=1;
for uu=2:size
    if jieguo(uu)<jieguo(mini)
        mini=uu;
    end
end
printresult(jieguo, mini, point, xx, Gender); %打印gender代结果
live=flive(size, point, jieguo, xx); %产生适应度（存活率）

```

```

new=zeros(1,size);      %根据适应度分配每个样本的选择次数
for k=1:size
    r(k)=rand;
end
for k=1:size
    j=1;
    if r(k)<=live(1) new(1)=new(1)+1;
    continue;
end
while r(k)>live(j)
    j=j+1;
end
new(j)=new(j)+1; %这样的分配保证了样本总和仍为size
end

bincode=zeros(size,point,6); %将第gender代的数据转换为二进制，准备交叉和变异
for k=1:size
    for j=1:point
        tmpo=xx(k,j);
        for kp=6:-1:1      %二进制逆序输出
            bincode(k,j,kp)=mod(tmpo,2);
            tmpo=floor(tmpo/2);
        end
    end
end

select=zeros(size,point,6); %依照new的个数更新size个样本
jishunew=1;
for j=1:size
    for k=1:new(j)
        for p=1:point
            for q=1:6
                select(jishunew,p,q)=bincode(j,p,q);
            end
        end
        jishunew=jishunew+1; %保证个数仍为size个
    end
end

cross=crosssample(size,point,select); %交叉
bianyi=trans(size,point,cross);      %变异
tmpp=zeros(size,point);              %将二进制重新转化会十进制
for k=1:size
    for p=1:point

```

```

        for q=1:6
            tmpp(k, p)=tmpp(k, p)+bianyi(k, p, q)*(2^(6-q));
        end
    end
end
for k=1:size                %±保证所有数据在1—51之内
    for j=1:point
        if tmpp(k, j)>51 || tmpp(k, j)<1
            tmpp(k, j)=randi(51);
        end
    end
end
for ip=1:size
zhiabiao=1;
    while(zhiabiao==1)
        zhiabiao=0;
        for j=1:point-1
            for k=j+1:point
                if tmpp(ip, j)==tmpp(ip, k)
                    zhiabiao=1;
                    tmpp(ip, j)=randi(51);
                end
            end
        end
    end
end
for g=1:size    %给新生成的组合按升序排序
    for j=1:point-1
        for k=j+1:point
            if tmpp(g, k)<tmpp(g, j)
                tmp=tmpp(g, k); tmpp(g, k)=tmpp(g, j); tmpp(g, j)=tmp;
            end
        end
    end
end
for k=1:size
    for j=1:point
        xx(k, j)=tmpp(k, j);
    end
end

Gender=Gender+1;
end

```

Create 函数:

```
function f=create(minn,maxn,n)
x=zeros(1,n);
same=1;
while same==1
for k=1:n;
x(k)=randi(maxn-minn+1)+minn-1;
end
same=0;
for k=1:n-1
    for j=k+1:n
        if x(j)==x(k)
            same=1;
        end
    end
end
end
for k=1:n-1
    for j=k+1:n
        if x(j)<x(k)
            t=x(j);x(j)=x(k);x(k)=t;
        end
    end
end
end
f=x;
end
```

计算函数:

```
function f=result(size,ys,data,point)
w=zeros(1,size);
wt=zeros(1,size);
for S=1:size
for k=1:400
for j=1:51
t=abs(ys(S,k,j)-data(k,j));
if t<=0.4 continue;
end
if t<=0.6
w(S)=w(S)+0.1;
continue;
end
if t<=0.8
```



```

w(S)=w(S)+0.7;
continue;
end
if t<=1
w(S)=w(S)+0.9;
continue;
end
if t<=2
w(S)=w(S)+1.5;
continue;
end
if t<=3
w(S)=w(S)+6;
continue;
end
if t<=5
w(S)=w(S)+12;
continue;
end
w(S)=w(S)+25;
end
end
for k=1:size
wt(k)=w(k)/400;
end
A=zeros(1,size);
cost=12*point;
for k=1:size
A(k)=wt(k)+cost;
end
f=A;
end

```

打印函数：

```

function printresult(A, index, point, xx, Gender)
fprintf(' Gender=%d cost:%f\n', Gender, A(index));
for j=1:point
fprintf(' %d ', xx(index, j));
end
fprintf('\n\n');

```

适应度函数：

```

function f=flive(size, point, A, xx)
for k=1:size
temp(k)=10000/(A(k)^3);
end
zong=0;
for k=1:size
for j=1:k-1
if temp(k)==temp(j)
    zhibiao=1;
    for g=1:point
        if xx(k, g)==xx(j, g)
            zhibiao=0;
            break;
        end
    end
    if zhibiao==1
        temp(k)=0;
        break;
    end
end
end
zong=zong+temp(k);
end
live(1)=temp(1)/zong;

for k=2:size
    live(k)=temp(k)/zong+live(k-1);

end;
f=live;
end

```

交叉函数：

```

function f=crosssample(size, point, select)
cross=select;
rd=randperm(size);
for k=1:2:size
    for p=1:point
        r=randi(6);
        for q=1:r
            cross(k, p, q)=select(rd(k), p, q);
            cross(k+1, p, q)=select(rd(k+1), p, q);
        end
    end
end

```

```

        for q=r+1:6
            cross(k, p, q)=select(rd(k+1), p, q);
            cross(k+1, p, q)=select(rd(k), p, q);
        end
    end
end
f=cross;
end

```

变异函数:

```

function f=trans(size, point, cross)
for k=1:size
    for j=1:point
        tmp=randi(100);
        if tmp==1
            a=4+randi(2);
            if cross(size, j, a)==1
                cross(size, j, a)=0;
            else
                cross(size, j, a)=1;
            end
        end
    end
end
f=cross;
end

```

老师的程序:

%%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%%

```

my_answer=[ 2, 10, 21, 30, 41, 49 ];%把你的选点组合填写在此
my_answer_n=size(my_answer, 2);

```

```

% 标准样本原始数据读入
minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample, npoint);
y0=zeros(nsample, npoint);
y1=zeros(nsample, npoint);
for i=1:nsample
    x(i, :)=minput(2*i-1, :);
    y0(i, :)=minput(2*i, :);

```

```

end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;

% 定标计算
index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % 请把你的定标计算方法写入函数mycurvefitting
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% 成本计算
Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-le1_0)+6*(
le3_0-
le2_0)+12*(le5_0-le3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果
fprintf('\n经计算，你的答案对应的总体成本为%.2f\n',cost);

function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];

% 将你的定标计算方法写成指令代码，以下样式仅供参考
y1=interp1(x_premea,y0_premea,x,'spline');

```

end