统计推断在数模转换系统中的应用

组号：13 姓名：沙荷茗 学号：5130309372，姓名：魏桐雨 学号 5130309679

摘要：本文主要研究如何使用统计推断的方法确定传感器部件检测的对象物理量 Y 与传感部件的输出电压信号用符号 X 的关系。采用插值与拟合的方法定量描绘 Y-X 曲线的形态，并且通过分析给定的标准样本数据库，使用启发式搜索算法给这个数模转换系统进行高效定标。在该研究问题背景下，比较了不同启发式算法搜索最佳解的能力，优化了算法参数。

关键词：统计推断，线性拟合，样条插值，数模转换系统定标，启发式搜索，遗传算法

1 引言

1.1 问题的提出

监测模块的组成框图如图 1-1。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）。

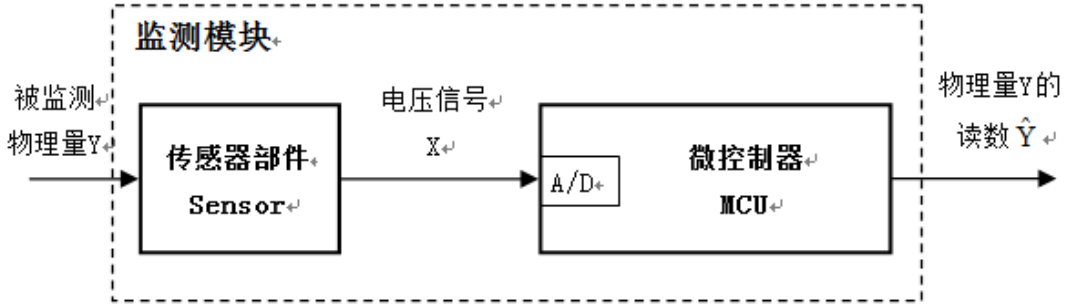

图 1-1 监测模块组成框图[1]

我们可以通过调节传感器可获得不同的输出电压 X，我们一共测量 51 组离散的数据。通过分析输入与输出的一个 X -Y 受控关系，来研究能否得到 X-Y 关系的一个曲线函数表达式。同时，在保证定标准确的前提下考虑能否减少测量点的个数，可以得到对整个样本空间适用的 X -Y 关系。如果将这个数模转换可调电源系统大规模生产，我们只需要测定几组电压就可以得到整条曲线的 X- Y 关系，达到节约人力物力的目的。

1.2 传感器部件特性

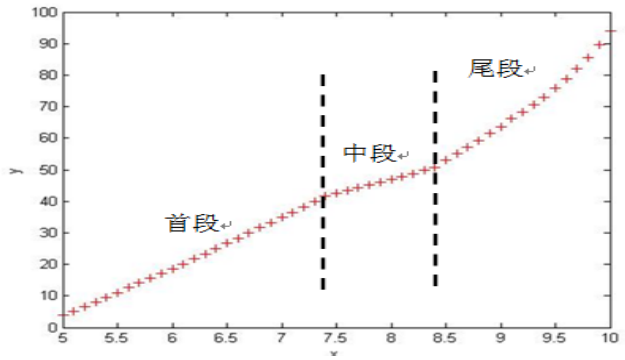

图 1-2 传感特性图示

一个传感部件个体的输入输出特性大致如图 1-2 所示，有以下主要特征[1]：

（1） Y 取值随 X 取值的增大而单调递增；

（2） X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；

（3） 不同个体的特性曲线形态相似但两两相异；

（4） 特性曲线按斜率变化大致可以区分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段；

（5） 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；

（6） 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 1-3 对比展示了四个不同样品个体的特性曲线图示。

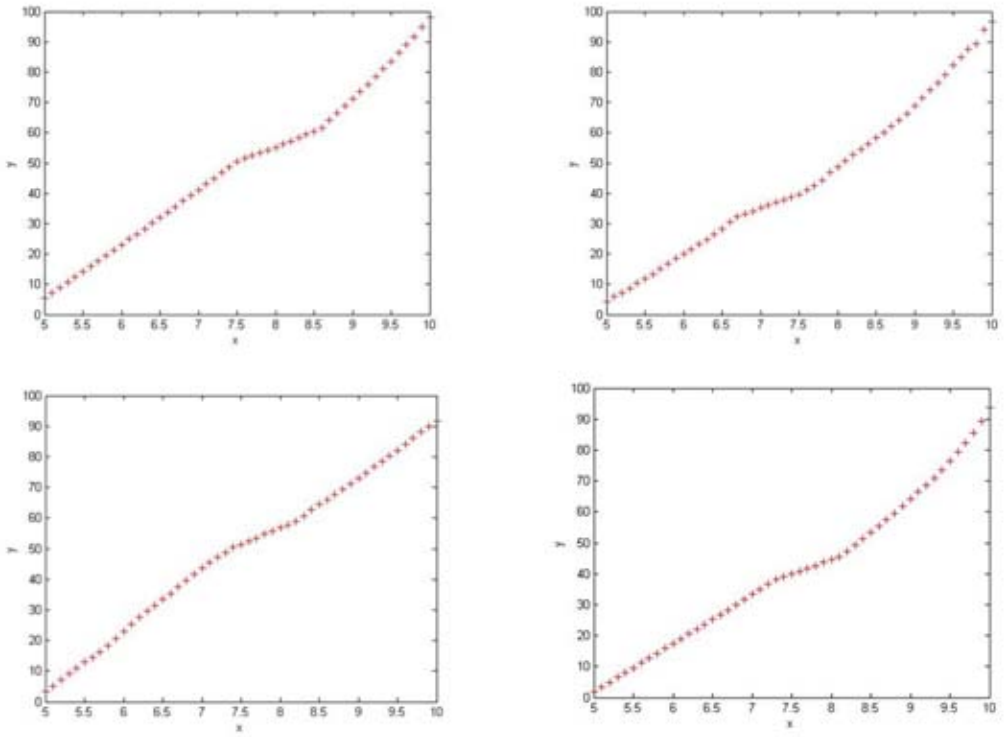

图 1-3 四个不同样本个体特性图示对比[1]

2 曲线拟合或插值方法的比较

2.1 提出假设

选择正确的拟合方式对于能否正确处理实验数据得到较为精确的结果有重要的意义。通常实验所采用的做法主要有以最小二乘法为原理的多项式拟合、指数对数函数拟合、傅立叶级数拟合以及插值计算。接下来我们大胆的假设曲线的形式为三次曲线，用实验数据来比较不同插值和拟合两种曲线表达的优劣。用实验数据证明三次曲线的表达效果优于其他形式的曲线。

2.2 选择曲线拟合方式

首先我们先尝试着从 469 组数据中任意选取 3 组 X- Y 数据，分别采用一次、二次、三次、四次曲线进行拟合，得到的结果类似。其中一组得到的结果如图 2 -2 所示

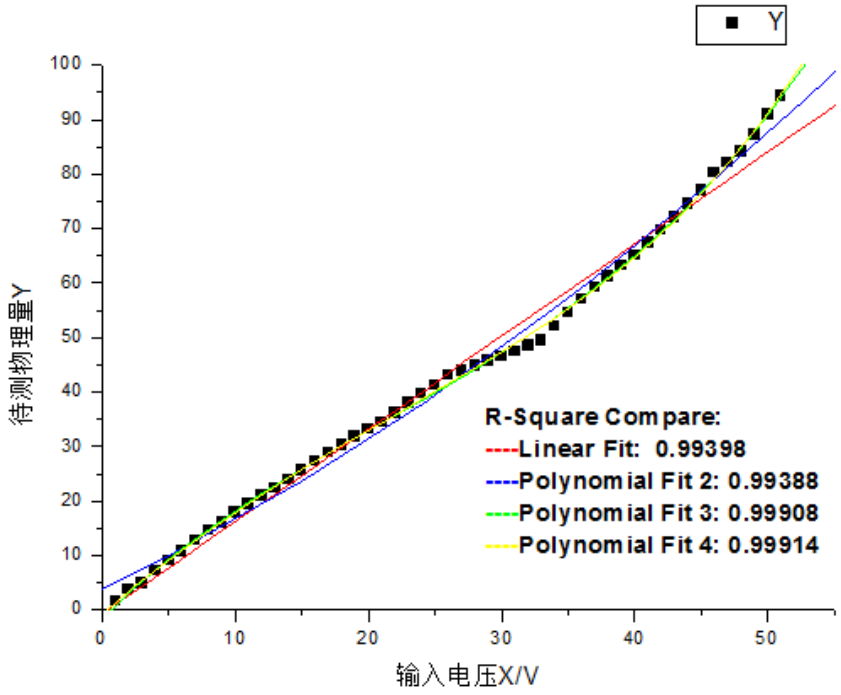

图 2-2 不同次数多项式 Y-X 曲线拟合效果比较

图形的趋势与三次曲线更为相似，而三次拟合出得曲线，与点看着更为贴切。但根据实际结果四次拟合的成本更低。

考虑到拟合曲线时，需要选取适当的特征点，如果直接使用 51 个数据点进行拟合会比较耗时，并且假设逐个从 469 组 51 个电压数据中选取 7 组特征点进行拟合观察，将会有 $51\times C^{7}_{51}\approx 1.15\times 10^{8}$ 也就是大约 1.15 亿种组合。对于 PC 机来说这将是一个天文数字般的运算，Matlab 运行需要很长时间。并且按照排列组合选取的七个点不一定都具有代表性，例如选取到相邻的七个点拟合的曲线在整个样本空间里是有一定偏差的。并且当前研究问题的主要矛盾是不同拟合方式选取优劣的比较，而不是如何选取最恰当的七个特征点，所以在实验中我们将随机均匀选取以下九组特征点：

(1)[1,11,19,28,35,45,51]； (2)[1,11,20,27,36,44,51]； (3)[1,12,18,28,35,46,51]
(4)[1,10,19,27,36,45,51]； (5)[1,13,17,26,34,44,51]； (6)[1,11,18,27,36,45,51]
(7)[1,12,17,27,36,44,51]； (8)[1,10,18,29,36,44,51]； (9)[1,10,19,27,34,45,51]

当然选取的这些特征点不一定是最佳的特征点，选取多个特征点是为了减少使用单组数据所带来的偶然性，从多组特征点数据来对不同拟合方式进行比较，得出的结论更有说服力。

2.2.1 多项式拟合方法

将测量结果分成三个区间进行，得到三段多项式拟合曲线的表达式：

区间 1 ： $Y_i = a_nX_i^n + a_{n-1}X_i^{n-1} + \cdots + a_1X_i + a_0 + \varepsilon_i$ （2-1）

区间 2 ： $Y_j = b_nX_j^n + b_{n-1}X_j^{n-1} + \cdots + b_1X_j + b_0 + \varepsilon_j$ （2-2）

区间 3 ： $Y_k = c_nX_k^n + c_{n-1}X_k^{n-1} + \cdots + c_1X_k + c_0 + \varepsilon_k$ （2-3）

其中： X 代表电压，Y 代表检测的物理量，n 代表次数， ε 代表残差

当前问题就是如何选取多项式拟合曲线的次数，使得拟合曲线最接近所有的数据。

我们使用 Matlab 对于选取的九组数据进行了计算，得到的结果如表 2 -1 所示：

表 2-1 不同拟合方式所对应的残差平方和的平均值

三次拟合	四次拟合	五次拟合	六次拟合
0.0174	0.0175	0.0186	0.0336

从表 2 -1 可以看出三次拟合与四次拟合效果最好不分伯仲，五次拟合的效果次之，而六次拟合的效果最差，残差的平方和也是明显最大的。其原因主要是出现了龙格现象，通常在用高阶多项式进行多项式插值时会出现次数越高而插值结果越偏离原函数的现象，因此我们当我们不熟悉曲线运动趋势的前提下，不要轻易使用高次插值。而使用分段多项式样条可以避免这个问题。如果要减小插值误差，那么可以增加构成样条的多项式的数目，而不必是增加多项式的阶次[2]。

2.2.2 样条插值拟合方法

随着问题研究的深入，我们思考能不能使用若干个离散的点的性质来表示整条曲线的性质。即只需要测量几个电压与占空比的点的数据，通过数学处理方法得到的曲线来代表整条数据空间中点所对应的关系，因而引出了样条插值法。

方法：我们在区间 [a,b] 上取 n+1 个节点 (n 为特征点的个数，这里我们取 7)

函数 y=f(x)在各个节点处的函数值为 S(x) ，若 S(x) 满足：

(1) $S(x_i)=y_i$ $i=0,1,\cdots,n$

(2) 在区间 [a,b] 上，S(x)具有连续的二阶导数；

(3) 在区间 $[x_i,x_{i+1}](i=0,1,\cdots,n-1)$ 上,S(x)是 x 三次的多项式；□

则称 是函数 y =f（x） 在 [a,b] 上的三次样条插值函数。

假定我们通过算法或者穷举法找出若干个能代表曲线的特征点，由三次样条插值多项式的性质我们知道必须每一段样条必须要有四个点才可以表示出中间两个点的一个函数关系，对于起始点和末尾点只需要三个点插值来表示曲线就可以了。

由 m 边界条件:$f'(a)=y_0$， $f'(a)=y_n$，即给定端点处的一阶导数值，建立起 g(x) 的完备三次样条插值函数。

表 2-2 多项式插值和样条插值残差平方和均值比较

三次多项式插值残差平方和均值	三次样条插值残差平方和均值
0.0174	0.0154

我们将不同曲线拟合的效果用七组特征点的样本残差平方和的平均值表示，结果如表 2-2 所示。我们可以看出三次样条插值的方法要比三次多项式插值的方法好，所以我们可以考虑使用三次样条插值的方法来得到曲线的表达形式。

3 寻找特征点为数模转换系统定标

3.1 成本计算

为评估和比较不同的校准方案，课程规定了以下成本计算规则[1]。

（1）单点定标误差成本

$$s_{i,j}=\begin{cases}0 & \text{if } |\hat{y}_{i,j}-y_{i,j}|\le 0.5\\ 0.5 & \text{if } 0.5<|\hat{y}_{i,j}-y_{i,j}|\le 1\\ 1.5 & \text{if } 1<|\hat{y}_{i,j}-y_{i,j}|\le 2\\ 6 & \text{if } 2<|\hat{y}_{i,j}-y_{i,j}|\le 3\\ 12 & \text{if } 3<|\hat{y}_{i,j}-y_{i,j}|\le 5\\ 25 & \text{if } |\hat{y}_{i,j}-y_{i,j}|>5\end{cases} \quad (3\text{-}1)$$

单点定标误差的成本按式（3-1）计算，其 $y_{i,j}$ 中表示第 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。

（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）某一样本个体的定标成本

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

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。

（4）校准方案总体成本

按式（3-2）计算评估校准方案的总体成本，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。

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

总体成本较低的校准方案，认定为较优方案。

3.2 特征点分布的定性分析

我们使用 Matlab 编写了一个对于现有给定特征点测定成本的程序，在开始尝试寻找特征点之前探探路，摸清大概情况，再设计具体的算法来挖掘出符合条件的特征点的序列。

3.2.1 定义观测平均成本

根据常识，指定首尾两点必取为测定点。我们知道去除首尾两个点之后还剩余中间 49 个输出电压数值，而且我们知道输出电压是均匀变化的，因此可以编号为 1~ 49 号。然后尝试任意选出若干个编号作为特征点，拟合后得到函数曲线，分别带入 1 ~ 49 号输出电压得到事后观测值 Yj,再与事前观测值 比较带入得到一个点的分数 Sj 相加之后得到一组观测值的总分 S 。但是因为每一组的观测的情况不同，总分 S 并不能代表所有观测的情况。所以使用 469 组观测平均成本来表示特征点与大量数据点的的吻合程度。

3.2.2 特征点分布情况

按照经验推断：选取的特征点应该在 1~51 号区间上均匀取点，平均成本才会比较低。接下来我们分别尝试不同的点的组合，大致了解平均成本的一个范围。假设我们就从 51 个点中任意选取 5 个点。下面是几种不同情况下的特征点以及平均成本。

(1）极端情况

点的分布过于密集，集中在某一个极小的区间内：

例：[2,3,4,5,6,]	平均成本：1085.28
例：[30,31,32,33,34]	平均成本：1046.59

可以看出得出的平均成本极高，是无法接受的。

(2）普通情况

点的分布不均匀，在某个区间内密集在另一个区间内分布疏散：

例：[3,20,43,45,49]	平均成本：147.96
例：[3,4,6,20,48]	平均成本：351.16

(3）优化情况

点在整个区间上分布均匀

例：[8,19,30,39,48]	平均得分：173.81
例：[9,20,29,41,49]	平均得分：192.38

从初步的试验特征点的情况来看，特征点的选取必须考虑到要在整个区间内均匀分布，

否则得到的成本会比较低。但是如何才能找出最合适的七个点，以及能否用更少点来表示整个区间的性质这些问题的回答需要借助“遗传算法”算法来寻找。

4 寻找特征点的算法

4.1 遗传算法的一些名词解释

(1)染色体：将样本的 51 个位点假设为 51 个基因分别位一个染色体上 1～51 号基因分别代表样本的 1～51 号横坐标。

(2)个体：包含 N 个互异的染色题的组合(2≤N≤51)每个个体至少包含 1 与 51 号基因（例如 a1=[1,2,3,4,5,6,51] a1 为一个个体初始种群)。含有 M 个个体的集合（M=50）初始个体的 N-2 个基因由随机函数生成。

(3)环境容量：在整个种群中的最多的个体的数目（max=100)。环境容量为一个标准实际情况个体个数有超过环境容量的情况出现。

(4)繁殖：对种群中的个体进行增值与变异。

(5)增值：将成本符合一定条件的个体集合中随机抽取出两个，随机让其 2*N-4（每个个体的一号与 N 号染色体不进行交换）个染色体随机组合，并平均分配到每个个体中。

(6)变异：个体的染色体(2^N-1 号)进行随机变异。

(7)种群平均成本：种群中所有个体种群成本的平均值。

(8)剪枝：当种群中的个体总量大于环境容量时，将每个成本大于种群平均成本的个体淘汰。

(9)遗传代数：进行剪枝的次数。

(10)退出条件：a.当种群中高于平均成本的个体数 N 占种群个数 M 的百分比小于收敛值

$$R(\frac{N}{M}\times 100\% < R)$$

b.剪枝次数大于最大剪纸次数。

4.2 算法流程

程序模拟生物定向育种的过程，将较为优秀的个体增值进化。算法流程如图 4-1。

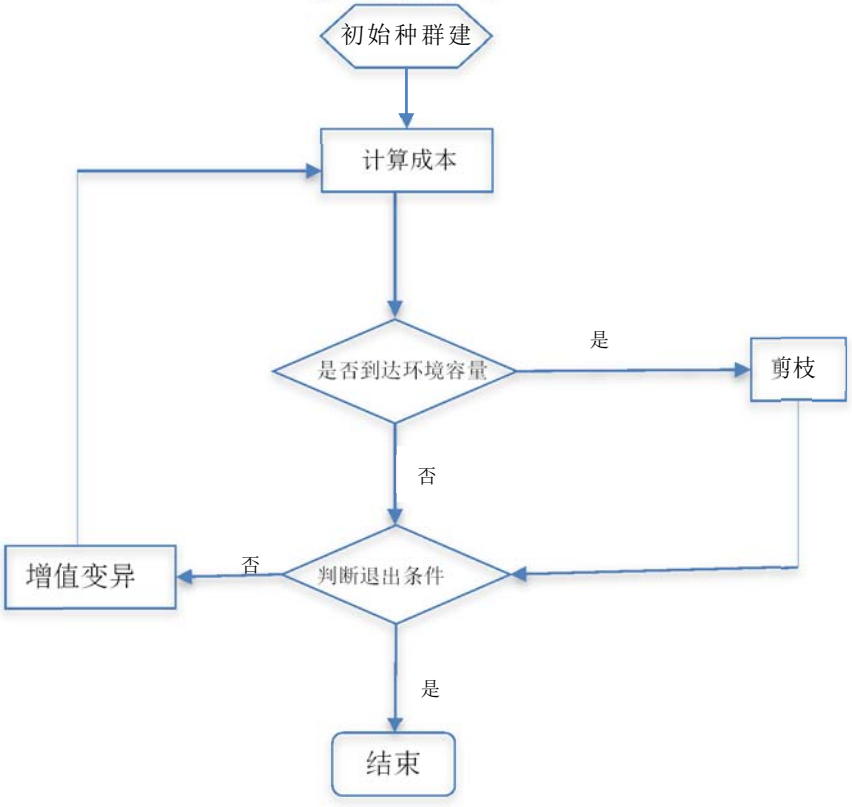

图 4-1 算法流程

4.3 算法分析

4.3.1 收敛性及可行性

我们不妨以取六个染色体的个体为例。

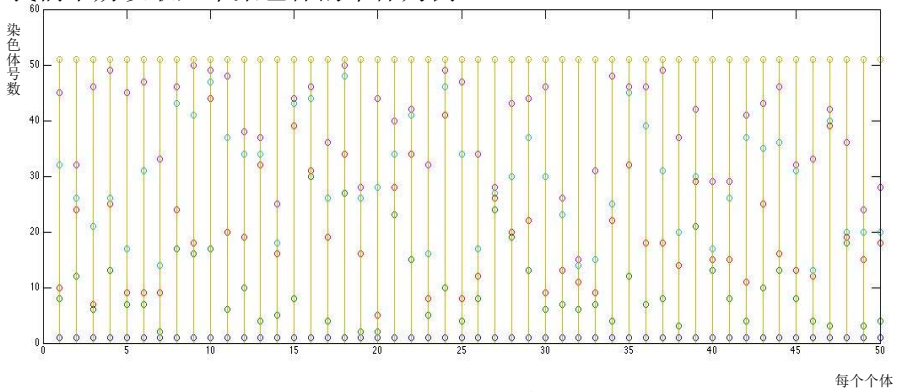

图 4-2 第一次随机生成的种群

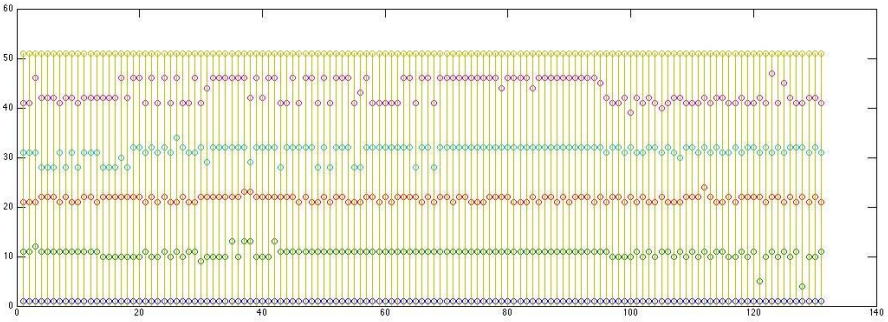

图 4-3 第五次剪枝后的个体情况

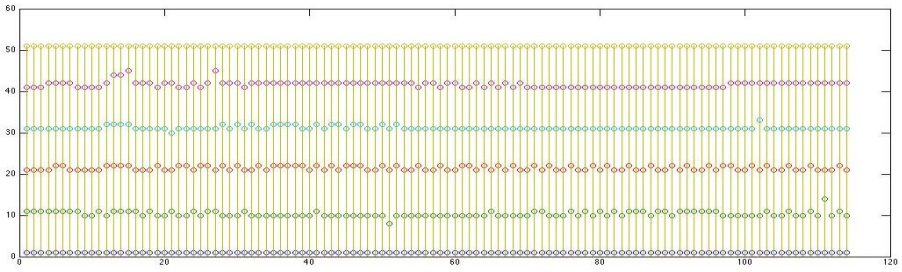

图 4-4 第十次剪枝后的个体情况

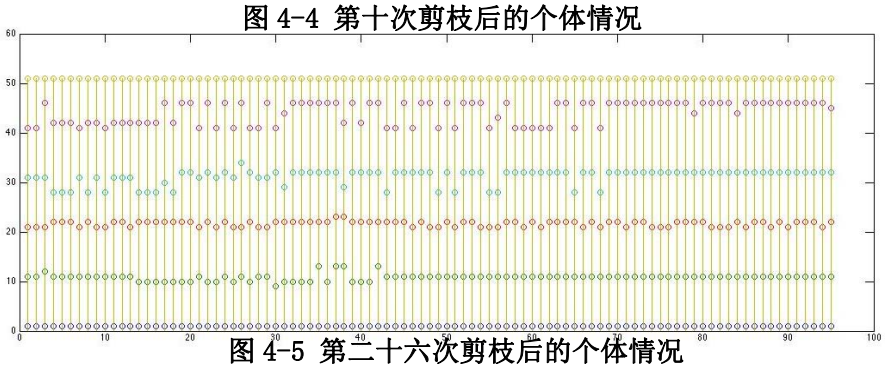

图 4-5 第二十六次剪枝后的个体情况

图 4-2 为随机生成的种群。可以发现数据分散且随机，但是在经过多次剪枝之后，数据趋于稳定，且均匀分布，这说明，最终获得的结果收敛性良好，算法可行性良好。第五次，第十次，第二十六次剪枝后的个体情况如图 4-3,4-4,4-5 所示。

4.3.2 关于不同选点个数对成本的影响

在对 7 个点的情况进行分析以后，我们对于增加特征点的个数能否对于成本有明显提高这个问题比较感兴趣。所以尝试着通过修改原来遗传算法的代码观察更多的特征点例如 4~9 个的结果。

表 4-1 不同选点个数对成本的影响

特征点序列	469 组数据平均成本	剪枝次数
[1 15 39 51]	137.6418	25
[1 12 26 39 51]	110.4488	25
[1 10 21 31 43 51]	95.1898	15
[1 9 20 28 35 44 51]	95.3710	32
[1 9 19 25 31 37 46 51]	102.7212	27
[1 5 14 20 26 32 38 46 51]	111.8102	23

从表 4-1 可以看出，比较特征点个数从 4 个到 9 个我们很清楚的看到在第 6,7 个点有最小值，而在大于 7 和小于 6 的情况下平均成本会越来越高。因而当特征点过少或者过多，都不能实现成本的低廉。所以最佳取点个数为 6,7 个点。

4.3.3 数据分析

从图 4-6 可以看出：

（1）在进化的前期种群的整体成本下降迅速，但是到达一定代数后成本下降趋于稳定

（2）在进化过程中会出成本现波动，这是由于进化的随意性决定但是总体的趋势是趋于减少。

（3）获得的最佳点位间隔与数据关系取相邻三点，有如下关系

$$(X_N - X_{N-3})/3 \propto df(X)\,(X = X_{N-2}) \qquad (4\text{-}1)$$

样本数据总体为线性关系，故最佳取点的间隔近似相等，所以运行结果符合条件。

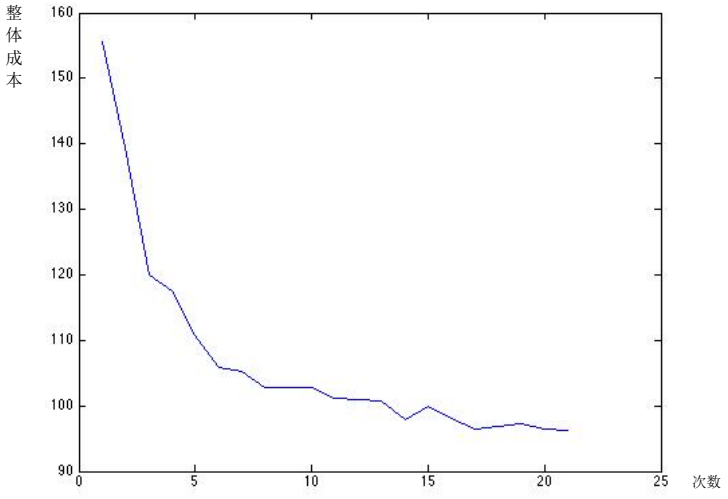

图 4-6 每次剪枝时的种群的平均成本

5 结论

由于遗传算法不能够遍历所有的可能解，在目前我们通过多次运行后得到的最好结果是 95.19，特征点序列为[1，10，20，28，35，44，51] 。我们不能否认找不到成本低于最好结果的特征点。但是我们发现使用三次线性拟合和三次样条插值的方法即使增加特征点个

数也不能使平均成本得到更大的降低，似乎分数存在一个极限值 95。如果需要降低成本，只有进一步优化算法或者是采用其他曲线的表达形式。

6 参考文献

[1] 上海交大电子工程系·统计推断在数模转换系统中的应用课程讲义 [EB/OL].ftp://202.120.39.248。

[2] 许小勇,太勇. 三次样条插值函数的构造与 Matlab 实现 云南民族大学数学与计算机科学学院，云南昆明

附页代码：

```
tic;
data=importdata('2014.csv');%读入数据
datax=data(1,:);
 ctime = datestr(now, 30);
 and('seed',ctime(13));
datay=data(2:2:end,:);%将数据读入表格 datax 放得是横坐标 datay 是每个样本的 y
data_s=size(datay);
sample_num=data_s(1,1);%样本的总数
RR=0.2;%收敛系数 值越小越收敛
rr=0;
u=0;
MAX_X=51;
change=0.4;%当变异率
MAX_G=40;%循环代数
generation=0;%记录循环代数
MAX_SIZE=100;%种群规模
a1=[1,3,4,5,6,7,11,26,51];%设置染色体数目样本个体增加或减少 2^（N-2）号染色体
a_s=size(a1);
measure_x=a_s(1,2);
% %  a1=[1,17,23,27,42,51];
%  a2=[1,3,4,10,21,45,51];
% a2=[1,10,22,31,43,51];
% a3=[1,25,26,36,45,51];
% a4=[1,10,21,31,43,51];
% a5=[1,11,21,31,41,51];
% a6=[1,10,22,31,43,51];
size_herd=50;
measure=zeros(MAX_SIZE*2,measure_x);
G_son=zeros(1,MAX_G);%每一代产生的孩子
g_parents=zeros(MAX_SIZE,measure_x);%记录可以遗传的父本
g_son=zeros(MAX_SIZE,measure_x);
s_g_son=0;
measure(1:end,1)=1;
measure(1:end,measure_x)=51;
 measure(1,:)=a1;
%  measure(2,:)=a2;
% measure(3,:)=a3;
% measure(4,:)=a4;
% measure(5,:)=a5;
% measure(6,:)=a6;
```

```
a=a1;

for i=2:50
    flage=1
    ctime = datestr(now, 30);
    and('seed',ctime(13));
    while(flage==1)

        for j=2:measure_x-1
            a(1,j)=ceil((MAX_X-2)*rand)+1;
        end
        a=sort(a);
        for j=2:measure_x
            if a(1,j)==a(1,j-1)
                flage=1 ; break;
            else
                flage=0;
            end
        end
    end
    measure(i,:)=a;
end

n=zeros(MAX_SIZE*2,1);
t_measure=zeros(MAX_SIZE*2,measure_x);
t_n=n;
sensor=[1:51];
size_sensor=size(sensor);
sensor_space=size_sensor(2);%测定点数
poses=51;
G_flage=1;%判断  是否遗传结束的标记
p_end=0.99;%判断收敛的系数
size_m=size(measure);%测定点数
rr_s=0;
current=1;
cost_count=[];

while(G_flage==1)

    for k=current:size_herd
        fit_x=datax(:,measure(k,:));%做为样本的抽样点（位置与个数应该调整）
        group_measure=zeros(1,sample_num);%放每一次的定标消费
        for i=1:sample_num
            sample_data=datay(i,:);%每次的 y 值
```

```
fit_y=sample_data(1,measure(k,:));%取出对于每个取值点得 测量值

fit_x=sort(fit_x);

fit_val=interp1(fit_x,fit_y,datax,'spline');

R=(fit_val-sample_data);

rr_s=rr_s+1;
fixed_cost=0;
for j=1:sensor_space;

    diference=abs(R(1,j));%测定|y^-y|

    if 5<diference
        fixed_cost=fixed_cost+25;

    else
        if 3<diference
            fixed_cost=fixed_cost+12;

        else
            if 2<diference
                fixed_cost=fixed_cost+6;

            else
                if 1<diference
                    fixed_cost=fixed_cost+1.5;



                else
                    if 0.5<diference
                        fixed_cost=fixed_cost+0.5;



                    end
                end
            end
        end
    end

end
```

```
        group_measure(1,i)=fixed_cost;

    end

    n(k,1)=sum(group_measure)/sample_num+12*measure_x;

end


%%%%%开始繁殖%%%%%%%%%%
g_parents(:,:)=0;%记录可以遗传的父本
s_g_parents=0;
G_AVERAGE=sum(n)/size_herd;

while MAX_SIZE<size_herd
    cost_count=[cost_count,G_AVERAGE];

    t_measure(:,:)=0;
    t_n(:,1)=0;
    i=0;
    for j=1:size_herd


        if  n(j,1)< G_AVERAGE
            i=i+1
            t_n(i,:)=n(j,:);
            t_measure(i,:)=measure(j,:);


        end

    end
    measure=t_measure;
    generation=generation+1;
    n=t_n;
    size_herd=i;
    G_AVERAGE=sum(n(:,1)/size_herd);

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%生完孩子 判断结束的条件，在将总体平均适应度即成本 平均值算出过后统计有多少大于他

rr= std([n(1:size_herd,1)]');

if rr<RR
    G_flage=0;
end

if MAX_G<generation
    G_flage=0;
end
g_count=0;
for i=1:size_herd;
    if n(i,1)<=G_AVERAGE

        g_count=g_count+1;
        g_parents(s_g_parents+1,:)=measure(i,:);
        s_g_parents=s_g_parents+1;

    end
end
u=g_count/size_herd;
if p_end<u
    G_flage=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if G_flage==1


    g_son(:,:)=0;
    s_g_son=0;
    ctime = datestr(now, 30);
    and('seed',ctime(13));
    %%%%%%%接下来是进行交配的函数


    i=1;
    j=1;
    while  s_g_son<s_g_parents
        while i==j
            if s_g_parents==1
                i=1;j=1;
```

```
            break;
        end
        i=fix(s_g_parents*rand+1);%随机选取两个进行交配 如果找到的是自己再循环
        j=fix(s_g_parents*rand+1);
    end

    flage=1;
    while(flage==1)

        p=fix(measure_x*rand)+1;
        g_son(s_g_son+1,:)=sort([g_parents(j,1:p-1),g_parents(i,p),g_parents(j,p+1:end)]);
        g_son(s_g_son+2,:)=sort([g_parents(i,1:p-1),g_parents(j,p),g_parents(i,p+1:end)]);
        for i_c=2:measure_x
            if g_son(s_g_son+2,i_c)==g_son(s_g_son+2,i_c-1)
                flage=1 ; break;

            else
                if g_son(s_g_son+1,i_c)==g_son(s_g_son+1,i_c-1)
                    flage=1 ; break;

                else
                    flage=0;
                end


            end
        end
    end

    s_g_son=s_g_son+2;
end

for i=1:s_g_son
    if change<rand

        flage=1;

        while(flage==1)
            a=g_son(i,:);
            P=ceil((measure_x-2)*rand)+1;
            a(1,P)=ceil((MAX_X-2)*rand)+1;
            a=sort(a);
            for i_c=2:measure_x
                if a(1,i_c)==a(1,i_c-1)
```

```
                        flage=1;break;
                    else
                        flage=0;
                    end
                end
            end


            g_son(i,:)=a;
        end

    end

    current=size_herd+1;
    measure(size_herd+1:size_herd+s_g_son,:)=g_son(1:s_g_son,:);%将孩子放进样本
    size_herd=size_herd+s_g_son;

end

end
```

```
[I,J]=min(n(1:size_herd,1));
out=[measure(J,1)];
i=0;
j=1;
for i=1:measure_x
    if out(j)~=measure(J,i)
        out=[out,measure(J,i)];
        j=j+1;
    end
end

disp('最小成本：');
disp(I-(measure_x-length(out))*20);
disp('选点个数：');
disp(length(out));
disp('选点方式：');
disp(out);
disp('遗传代数：');
```

```
disp(generation);

toc;
```