

# 统计推断在数模转换系统中的应用

组号 32 李智康 5130309326, 付浩 5130309325

**摘要：**通过统计推断，确定传感器  $x$ ,  $y$  之间的函数关系，为生产应用提供指导

**关键词：**AD 转换，统计推断，遗传算法

## 1 引言

在工程实践与科学实验中，我们通常面对的对象是一个由多个变量组成的一个系统。我们通常要研究的就是这些变量之间的函数关系，通常我们遇到最多的是两个变量的系统，一个输入变量，一个输出变量。在理论中我们常常可以通过计算得出两个变量之间的确定的函数关系。但在实际中，我们很难找到非常精确描述两个变量的函数，但是我们可以通过多种不同曲线拟合的方法算出多种不同的方案，最终比较每个方案的优劣程度来选出最优的方案。

### 1.1 课题概述

假定有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。本课题要求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。

#### 1.1.1 典型样品曲线特性分析

一个传感部件个体的输入输出特性大致如图所示，有以下主要特征：

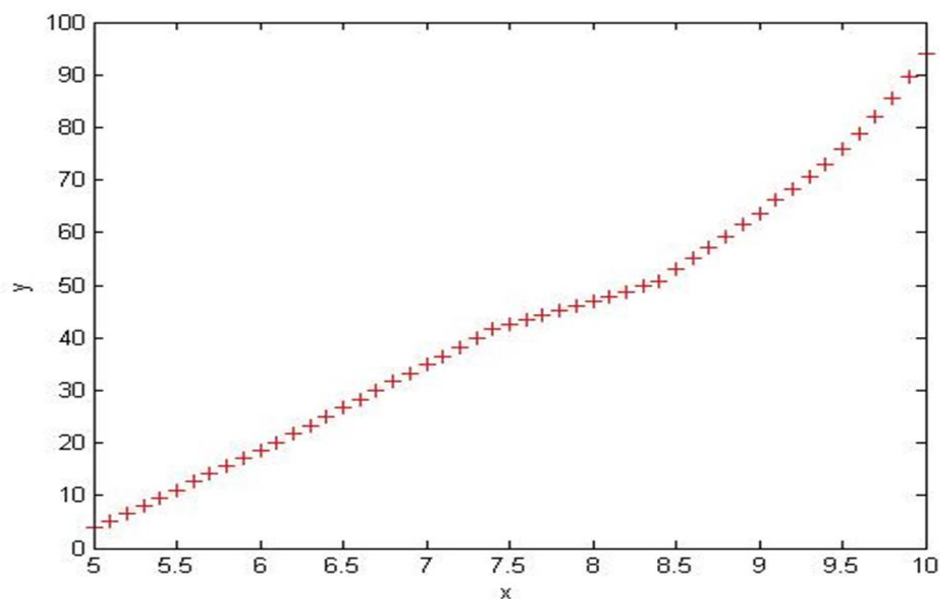


图 1 传感部件输入输出特性曲线

- 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；
- 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。

## 2 传感器校准的数值模拟方法

### 2.1 设计思路

本课程要完成一个最优（成本最低）的定标方案，因此

总成本分为测量成本和误差成本两个部分，当测定点数减少时，其余点要通过插值来获得，必然会增大误差成本，而测定点增多时，在减小误差成本的同时又会增大测定成本，因此要在两者之间找到平衡，使总成本最小。

因为样本的线性性态比较好，选用基本的表达式进行拟合（三次样条插值）。

对于相同单点测定次数  $n$ ，X 的组合方案有  $C_{51}^n$  种，是典型的 NP-hard 问题，运用遗传算法确定点的组合，从而有效的减少备选方案，快速得到近似解。

### 2.2 数学方法

#### 2.2.1 曲线拟合

用连续曲线近似地刻画或比拟平面上离散点组所表示的坐标之间的函数关系的一种数据处理方法。用解析表达式逼近离散数据的一种方法。在科学实验或社会活动中，通过实验或观测得到量  $x$  与  $y$  的一组数据对  $(x_i, y_i)(i=1, 2, \dots, m)$ ，其中各  $x_i$  是彼此不同的。人们希望用一类与数据的背景材料规律相适应的解析表达式， $y = f(x, c)$  来反映量  $x$  与  $y$  之间的依赖关系，即在一定意义下“最佳”地逼近或拟合已知数据。 $y = f(x, c)$  常称作拟合模型，式中  $c = (c_1, c_2, \dots, c_n)$  是一些待定参数。

最常用的一种做法是选择参数  $c$  使得拟合模型与实际观测值在各点的残差（或离差） $e_k = y_k - f(x_k, c)$  的加权平方和达到最小，此时所求曲线称作在加权最小二乘意义下对数据的拟合曲线。

#### 2.2.2 遗传算法

生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。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，不需要确定的规则。

### 3 具体方法的程序实现

#### 3.1 程序时间复杂度的分析

采用实验法，在相同条件下，经过测定各函数的运行时间，发现主要耗时在计算总成本的函数 `geney`。

以下为各主要函数的耗时（运行 100 次的测算）。

<code>initial</code> (初始化)	<1s
<code>ranky</code> (根据成本计算适应度)	<1s
<code>select</code> (选择)	<1s
<code>recombin</code> (重组)	<1s
<code>mut</code> (变异)	<1s
<code>geney</code> (计算成本)	30s

和 `geney` 相比，其它函数耗时可以忽略不计。因此以下所有降低耗时的方式本质上均为减少调用 `geney` 的次数，即减少收敛所用代数

#### 3.2 遗传算法的模型的构建

物理结构：采  $N \times 49$  矩阵存储种群，每一行为一个个体，每一列的序号  $k$  对应着数据样本中采样点的编号  $k-1$ （因为默认必须测量第一个点和最后一个点，所以将第二个点放在第一列）， $a_{i,j}$  的取值为 0 或 1，1 代表测量该点，0 代表不测量该点。（因为采用内插值必须要有首末点）。

因此，每个个体代表了一种选点方案， $N$  决定了种群规模。

遗传收敛的判据： $|\min(prices) - \text{average}(prices)| \leq 5$  且  $\min(prices) \leq 100$

因为考察选点方案优劣的唯一指标是总成本。因此将  $|\min(prices) - \text{average}(prices)| \leq 5$  作为可结束的判据。为了避免陷入局部最优后自动终止，增加判据  $\min(prices) \leq 100$ 。

整个运行过程中，不需要固定选点个数，依靠遗传算法自行进行选择。

#### 3.3 实现方法

见图 2

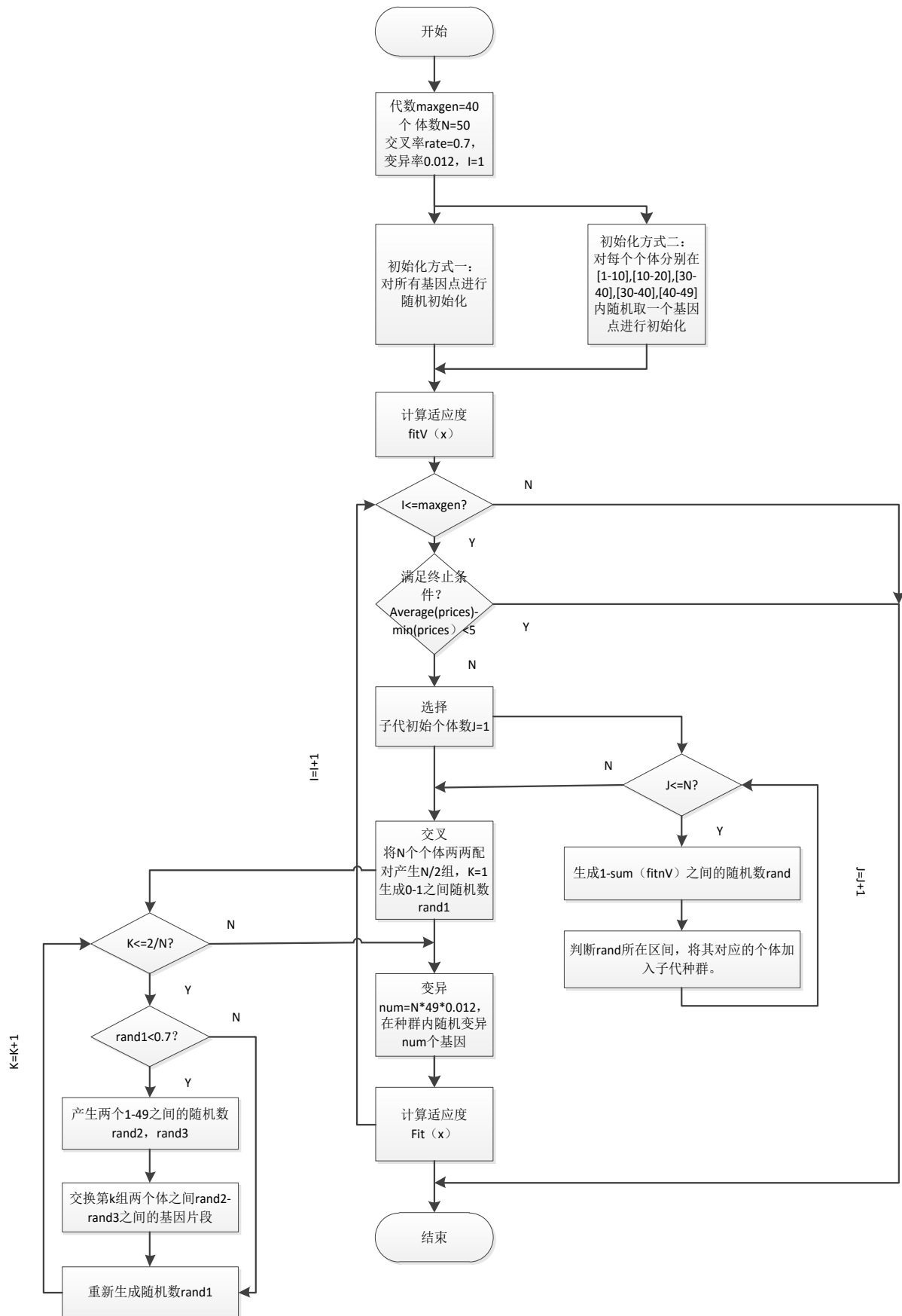


图 2 问题求解流程图

### 3.4 解释

#### 3.4.1 生成初始种群

使用两种方式：

1. 第一次使用，直接对整个矩阵生成随机数
2. 区间[0-10][10-20][20-30][30-40][40-50]中每个区间取一个点

解释：几次实验之后，发现测量点步长大致相同，整体呈均匀分布。且总体点数收敛为5个，因此将点分成5个区间。

这样可以减少初始的不必要个体，加快搜索速度，从而在相等时间成本下，扩大初始种群规模，覆盖更多可能的近似最优解。进一步避免收敛到局部最优解。

下表为一次结果的比较，（一次结果中）分别用方法1和2进行生成100个个体收敛所需的代数及时间：

表 3-1 不同初始化方式对实验结果的影响（其它变量均取默认条件）

初始化方式	初始种群规模	收敛代数	运行时间	选点方案	对应成本
1	100	12	13 分 42 秒	1 9 21 30 41 51	96.3
2	100	5	4 分 31 秒	1 9 19 28 35 44 51	95.71

#### 3.4.2 适应度函数（基于总成本 $prices[N]$ ）（ $fitnV$ ）

$$fitnV(x) = \begin{cases} (prices(x) - average(prices))^2, & prices(x) \leq average(prices) \\ 0, & prices(x) > average(prices) \end{cases} \quad (3-1)$$

因为总成本是越小越好，而适应数是越大越好。因此要完成两者之间的一种转换。为了加快优化的产生，对成本小的个体适应度进行了放大  
当个体  $x$  的成本大于种群的成本平均值时，适应度为 0，即不会产生子代（认为必定被淘汰）  
当个体  $x$  的成本小于种群成本平均值时，用种群成本平均值减去当前个体的成本后再平方即可，（当三次时，种群陷入局部最优解的几率增大，二次即可）这样增大了低成本（优秀）个体产生下一代的几率。

由下表一次试验  $fitnV$  与  $fitnV' = 1 / prices(x)$  的性能比较，可以清楚的看出本实验适应度的取法比起直接对误差取倒数的方式在加快搜索速度，降低运行时间方面的优势

表 3-2 不同适应度函数对实验结果的影响（其它变量均取默认条件）

适应度函数	初始种群规模	收敛代数	运行时间	选点方案	对应成本
$fitnV$	50	36	14 分 31 秒	1 10 21 28 36 42 51	99.15
$fitnV'$	50	47	20 分 11 秒	1 14 29 41 48 51	135.94

### 3.4.3 选择

采用赌轮算子思想，令  $a = \text{sum}(\text{fitnV})$  适应度函数之和。

产生  $N$  个  $(1-a)$  之间的随机数  $\text{rand}$ ，通过  $\text{rand}$  的大小确定能够遗传到下一代的父代。 $1-a$  可分为  $N$  段：

$$\{1 - \text{fitnV}(1)\}, \{\text{fitnV}(1) - \text{fitnV}(2)\}, \dots, \{\text{fitnV}(n-1) - a\}$$

当  $r_N$  位于  $\{\text{fitnV}(1) - \text{fitnV}(2)\}$  时，选择第二个父代  $x_2$ ，依次类推。

### 3.4.4 交叉

采用两点交叉，通过交叉率来  $\text{rate}$  控制

从上到下个体两两配对，如果有奇数个个体，则最后一个个体不参与交叉

对每对个体组，先生成一个 0-1 的随机数  $\text{rand}_1$ ，如果  $\text{rand}_1 < \text{rate}$ ，则进行交叉。

如果  $\text{rand}_1 \geq \text{rate}$ ，则不进行交叉

对进行交叉的个体组（个体 A，B），生成两个 1-49 的随机数  $\text{rand}_2, \text{rand}_3$ ，交换 A，

B 中  $\text{rand}_2, \text{rand}_3$  之间的部分。

由下表可知，两点交叉相对单点交叉并没有在搜索能力方面有显著的提高

表 3-3 不同交叉方式对实验结果的影响（其它变量均取默认条件）

交叉方式	初始种群规模	收敛代数	运行时间	选点方案	对应成本
两点交叉	50	36	14 分 31 秒	1 10 21 28 36 42 51	99.15
单点交叉	50	33	14 分 18 秒	1 10 21 26 34 43 51	95.95

### 3.4.5 变异

变异率为整个种群中变异基因的个数，本实验定为  $0.7/49=0.012$ ；

一般变异率取为 0.01-0.2，过小产生的新个体少，过大变成随机搜索。根据资料一般取  $0.7/\text{lind}$ （ $\text{lind}$  为染色体长度），因此本实验采用此法计算得到变异率。[4]

### 3.4.6 终止条件

达到固定代数  $N_{\text{ind}}$  或满足收敛判据，详见遗传算法模型的构建。

### 3.4.7 拟合方式

采用三次样条插值，用于计算成本  $\text{prices}$ 。而由前文述，运行时间主要取决于成本计算

函数 `geney`，所以拟合方式的好坏直接决定了每代所花的时间。经过实验，选取三次样条插值。下表为一次三次样条插值和三次多项式拟合运行的结果

表 3-4 不同拟合方式对实验结果的影响（其它变量均取默认条件）

拟合方式	初始种群规模	收敛代数	运行时间	选点方案	对应成本
三次样条插值	50	36	14 分 31 秒	1 10 21 28 36 42 51	99.15
三次拟合	50	68	30 分 48 秒	1 14 28 41 51	124.89

## 4 结论

取点为 5-7 个，各相邻点间步长大致均匀，使用三次样条插值时，总成本较小，在 95-100 之间。

样例取点方式 1, 11, 21, 31, 41, 51 成本 96.1。

## 5 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] Matlab 工具箱应用 苏金明编著  
[3] 蔡钧 基于遗传算法的测量数据拟合 扬州大学学报 007 824X(2002)04 0059 04  
[4] Andrew Chipperfield Genetic Algorithm TOOLBOX

## 附录(程序清单)

### **main.m**

%主函数

```
NIND=50;          %个体数目
MAXGEN=100;       %最大遗传代数
low=200;
diff=100;
```

%初始种群方式一

```
Chrom=round(rand(NIND,PRECI));
Lind=PRECI;
```

%初始种群方式二

```
% Chrom=zeros(NIND,PRECI);
% Lind=PRECI;
% for j=1:PRECI;
%   a=round(8*rand(1,5))+1;
%   for i=1:5
%       Chrom(j,10*(i-1)+a(i))=1;
%   end
% end
% Lind=PRECI;
```

```
gen=0;            %代计数器
ObjV=geney(Chrom,NIND);%成本计算函数
```

```
while gen<MAXGEN&&diff>5&&low>100
```

```
    FitnV=ranky(ObjV);          %分配适应度值
    SelCh=select(Chrom, FitnV); %选择
    SelCh=recombin('xovsp', SelCh, 0.7); %交叉
    SelCh=mut(SelCh);          %变异
    [r,c]=size(SelCh);
    ObjVSel=geney(SelCh,r);
    Chrom=SelCh;
    ObjV=ObjVSel;
    fprintf('次数%.5f,gen);
    fprintf('\n 经计算，你的答案对应的最小成本为%.5f\n',min(ObjV));
    fprintf('\n 经计算，你的答案对应的平均成本为%.5f\n',average(ObjV));
    gen=gen+1;
    diff=average(objV)-min(objV);
    low=min(objV);
```



end

%输出最终的选点方案

data=display(ObjV);

**function FitnV=ranky(ObjV)**

%适应度函数

%适应度函数一

a=average(ObjV)-ObjV;

a(a<0)=0;

FitnV=a.^2;

%适应度函数二

%FitnV=1./Obj;

**function SelCh=select(Chrom, FitnV)**

%选择

[Nind,ans] = size(FitnV);

cumfit = cumsum(FitnV);

trials = cumfit(Nind) .\* rand(Chrom, 1);

M1 = cumfit(:, ones(1, Chrom));

M2 = trials(:, ones(1, Chrom));

[NewChrIx, ans] = find(M2 < M1 &[ zeros(1, Chrom); M1(1:Nind-1, :) ] <= M2);

**function SelCh=recombin(Sel, rate)**

%交叉

%两点交叉

SelCh=Sel;

[row,col]=size(Sel);

border=rate\*10;

index=round((border-1)\*rand(1,col/2));

exchange=round(48\*rand(col/2,2));

for i=1:col/2

if index(i)<border

a=SelCh(i\*2-1,:);

b=SelCh(i\*2,:);

SelCh(i\*2-1,exchange(i,1):exchange(i,2))=b(1,exchange(i,1):exchange(i,2));

SelCh(i\*2,exchange(i,1):exchange(i,2))=a(1,exchange(i,1):exchange(i,2));

end

end

%单点交叉

% exchange=round(48\*rand(col/2,1));

% for i=1:col/2

% if index(i)<border

% a=SelCh(i\*2-1,:);

% b=SelCh(i\*2,:);

% SelCh(i\*2-1,exchange(i,1):49)=b(1,exchange(i,1):49);

% SelCh(i\*2,exchange(i,1):49)=a(1,exchange(i,1):49);

% end

% end

**function SelCh=mut(Sel);**

%变异

num=45;

a=round(rand(1,num));

b=round(469\*51\*rand(1,num));%生成变异点

temp=1-Sel(b);

SelCh=Sel;

**function ObjV=geney1(Chrom,NIND)**

%求种群的成本

data(NIND,49)=0;

k(1,NIND)=0;

%检索选取点

for i=1:NIND

for j=1:49

if(Chrom(i,j)==1)

k(i)=k(i)+1;

data(i,k(i))=j+1;%因为 1 和 51 默认有，所以 1 对应的是二号点

end

end

end

%计算成本

ObjV(NIND,1)=0;

for i=1:NIND

kk=[1,data(i,1:k(i)),51];

```

ObjV(i,1)=prices(kk);
clear kk;
end

```

### **function cost=prices(my\_answer)**

%成本计算函数

```

my_answer_n=size(my_answer,2);

% 标准样本原始数据读入
minput=dlmread('20141010dataform.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
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% 成本计算
Q=12;
errabs=abs(y0-y1);
le0_5=(errabs<=0.5);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);
sij=0.5*(le1_0-le0_5)+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;
```

```
function y1 = mycurvefitting( x_premea,y0_premea )
```

```
%拟合函数
```

```
x=[5.0:0.1:10.0];
```

```
%三次样条插值
```

```
y1=interp1(x_premea,y0_premea,x,'spline');
```

```
y1=polyfit(x_premea,y0_premea,x,3)
```

```
%三次多项式拟合
```

```
% p=polyfit( x_premea,y0_premea,3);
```

```
%y1=polyval(p,x);
```

```
function data=display(ObjV);
```

```
%输出最终选点方案
```

```
%种群个体和实际选点的转换
```

```
data(NIND,49)=0;
```

```
k(1,NIND)=0;
```

```
for i=1:NIND
```

```
    if ObjV(i)<100
```

```
        for j=1:49
```

```
            if(Chrom(i,j)==1)
```

```
                k(i)=k(i)+1;
```

```
                data(i,k(i))=j+1;%因为 1 和 51 默认有，所以 1 对应的是二号点
```

```
            end
```

```
        end
```

```
    end
```

```
end
```