

# 统计推断在数模转换系统中的应用

组号：69 余惠婷（5140309147） 赵雅雯（5140309123）

**摘要：**本文在三次插条拟合法、遗传算法的基础上，介绍如何处理大量的电子产品数据并得到最优解的方法，从而合理高效地评估该电子产品的特性。

**关键词：**遗传算法，三次插条拟合法，定标

## Application of Statistical Inference in Analog-to-Digital Converting System

**Abstract:** Based on the genetic algorithm and the cubic spline interpolation, in this paper, there is an introduction about how to deal with large amount of data from the specifickind of electronic device to assess the performance of the electronic devicereasonably and efficiently.

**Keywords:** genetic algorithm, simulated annealing algorithm, cubic spline interpolation, calibration

### 1 引言

据本课程规定，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。

### 2 问题的来源和解决方案

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。

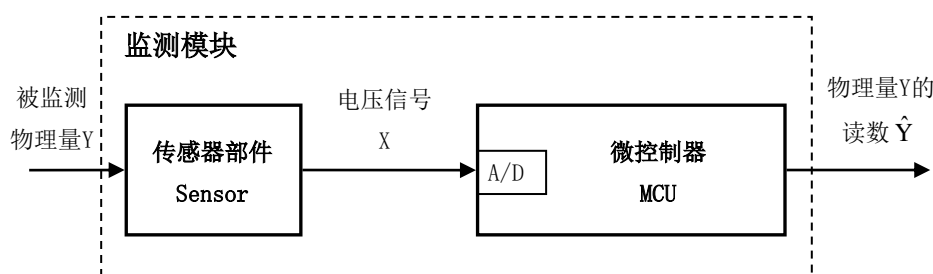


图 1 监测模块组成框图

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  $y_i$ ， $i = 1, 2, 3, \dots, 50, 51$ 。

## 2.1 传感部件特性

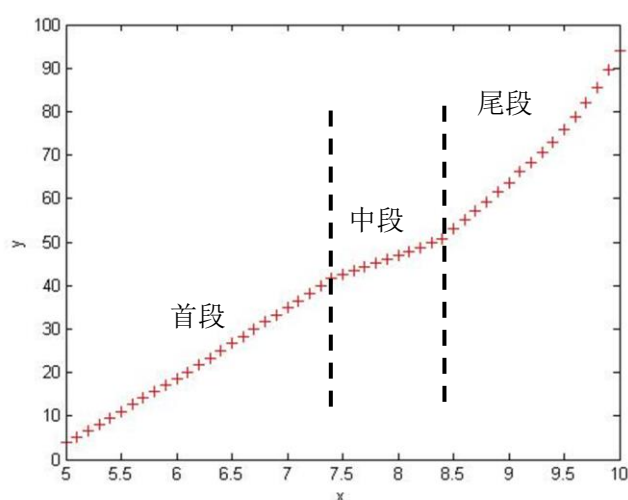


图 2 传感特性图示

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。

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。

## 2.2 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。

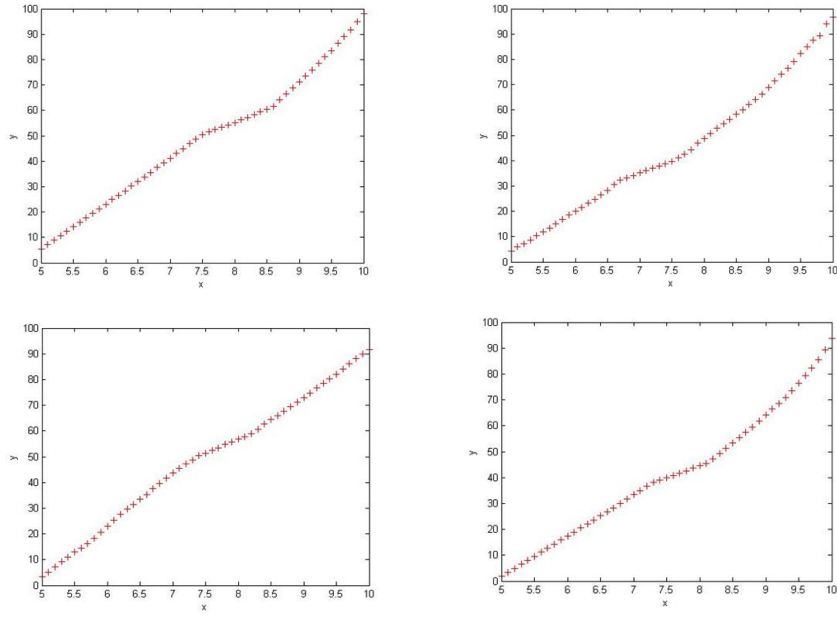


图 3 四个不同样本个体特性图示对比

## 2.3 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。

## 2.4 选点方法的选取

### 2.4.1. 穷举法

对于这个问题, 如果要测量所有的测试点必定要耗费大量的成本, 若是选取其中几个点, 再取遍所有情况 (以 7 点为例, 就要有 115775100 种情况), 很明显由于数据庞大, 要比较全部的方案, 成本太高, 穷举法不可取。

### 2.4.2. 遗传算法

遗传算法 (GA) 是一类借鉴生物进化规律形成的随机化搜索方法。遗传算法的实现可分为以下几个步骤: 首先随机建立初始状态即个体基因第一代, 根据实际问题求解函数来设定适应度函数。其次在接下来的若干代中, 依次进行: 每个个体适应度函数的求取, 根据适应度值的大小进行轮盘概率求取, 个体间进行交叉和变异, 繁殖出具有适应度更好的下一代。

遗传算法是解决搜索问题的一种通用算法, 对于各种通用问题都可以使用。搜索算法的共同特征为:

- (1) 首先组成一组候选解;
- (2) 依据某些适应性条件测算这些候选解的适应度;
- (3) 根据适应度保留某些候选解, 放弃其他候选解;
- (4) 对保留的候选解进行某些操作, 生成新的候选解。

在遗传算法中, 上述几个特征以一种特殊的方式组合在一起, 基于染色体群的并行搜索, 带有猜测性质的选择操作、交换操作和突变操作。这种特殊的组合方式将遗传算法与其它搜索算法区别开来。

遗传算法为循环处理数据提高准确度提供了便利, 同时避免了成本过大的问题, 值得考虑为处理问题的主要方法。

## 2.5 遗传算法的实现

### 2.5.1 遗传因素值的选取

- 1) 当种群大小<sup>1</sup>取值较小时, 可提高遗传算法的运算速度, 但却降低种群的多样性, 容易引起遗传算法早熟, 出现假收敛; 而当种群大小取值较大时, 又会使得遗传算法效率降低。因而本实验中种群大小取150。
- 2) 交叉概率<sup>2</sup>较大时, 程序运行时间相对较短, 但查阅资料得知: 交叉概率若过大, 可能破坏群体

---

<sup>1</sup> 见附录 main.m 第 13 行引入的 S

<sup>2</sup> 见附录 crossover.m 函数

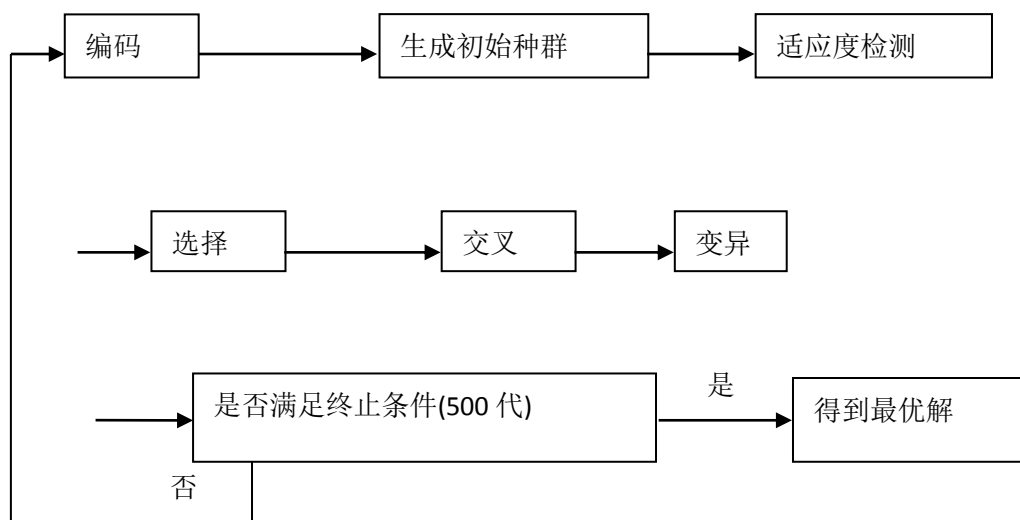
的优良模式。因而本实验中交叉概率取0.9。

- 3) 变异概率<sup>3</sup>小，产生新个体少；变异概率太大，又会使遗传算法变成随机搜索，因而本实验中变异概率取0.05。

在本次定标操作中，主要实现了对定标点和个数的选择，以150个个体为一个种群，每一个个体有51个基因<sup>4</sup>。采用轮盘赌的选择方法选择，采取倒数的方法令其为适应度，以0.05为变异概率，0.9为交叉概率，遗传代数设置为500代<sup>5</sup>。

## 2.5.2 遗传算法的主要思想和实现方式

- 1) 编码<sup>6</sup>：使用二进制编码，随机产生一个初始种群。本次编码为长度为51的基因链，其中0代表不选取此点，1代表选取此点作为定标点。
- 2) 生成初始种群：种群规模表示每一代种群中所含个体数目。
- 3) 适应度检测：根据实际标准计算个体的适应度，评判个体的优劣；
- 4) 选择<sup>7</sup>：从当前群体中选择适应度高的个体，使它们有机会被选中进入下一次迭代过程，舍弃适应度低的个体。本例中采用轮盘赌的选择方法，即个体被选择的几率与其适应度值大小成正比；而由于本次试验要求成本越小越好，所以采取倒数的方法令其为适应度。
- 5) 交叉<sup>8</sup>：遗传算法中最主要的操作，交叉概率较大时，程序运行时间相对较短，但查阅资料得知：交叉概率若过大，可能破坏群体的优良模式。因而本实验中交叉概率取0.9。
- 6) 变异<sup>9</sup>：随机选择中间群体中的某个个体，以变异概率大小改变个体某位基因的值。变异为产生新个体提供了机会。本实验中变异概率取0.05。
- 7) 终止进化代数：本例中规定遗传代数的收敛判据，根据设置好的进化代数，搜索到规定代数后自动停止，本次试验中设为500次。



图四 遗传算法程序流程图

<sup>3</sup> 见附录 mutate.m 第4行引入的0.05为比较边界

<sup>4</sup> 见附录 main.m 第22行引入的 Gene

<sup>5</sup> 见附录 main.m 第14行引入的 G

<sup>6</sup> 见附录 main 函数 22 行

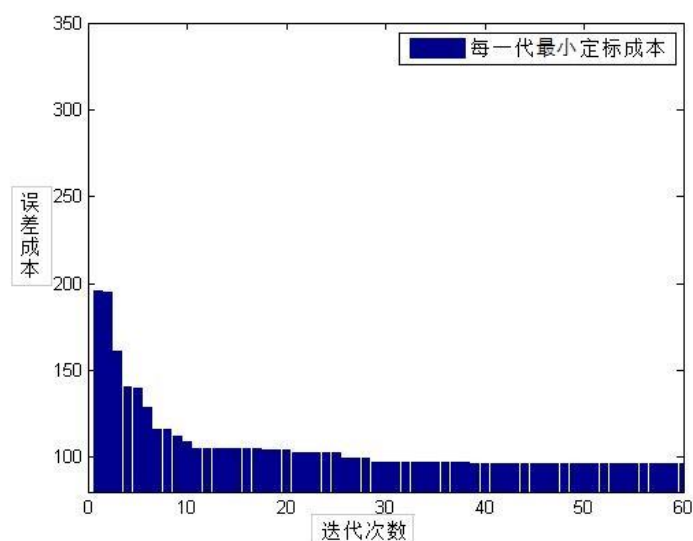
<sup>7</sup> 见附录 selection.m 函数

<sup>8</sup> 见附录 crossover.m 函数

<sup>9</sup> 见 mutate.m 函数

## 4 结果分析

### 4.1 运行结果



图五 遗传算法运行结果

表一 不同代数对应的取点位置和成本（运行时间：461s）

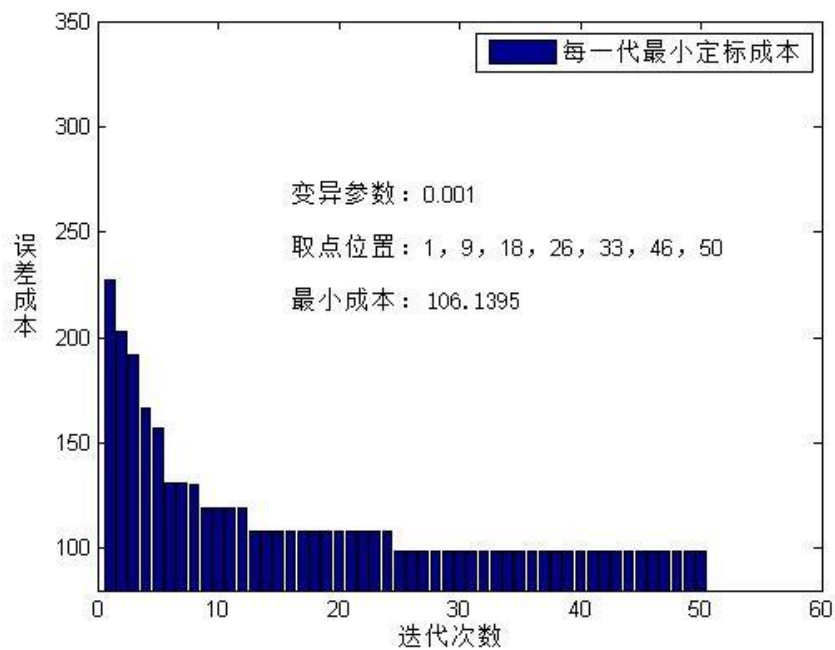
代数	取点位置	成本
47	2,11,22,28,35,44,50	96.5362
49	2,11,19,28,35,44,50	96.4472
59	2,11,21,28,35,44,50	95.8990
87	2,9,21,28,35,44,50	95.7963
112	2,9,20,28,35,44,50	95.5613
125	2,9,20,27,35,44,50	95.2852
209	2,9,20,27,34,44,50	95.2147

在运用三次插条拟合法、遗传算法时，运行代数设为 500 代，种群大小为 150，遗传概率为 0.05，所得最小成本为 95.2147，最优取点方案为 2,9,20,27,34,44,50，程序总共运行时长为 461s。

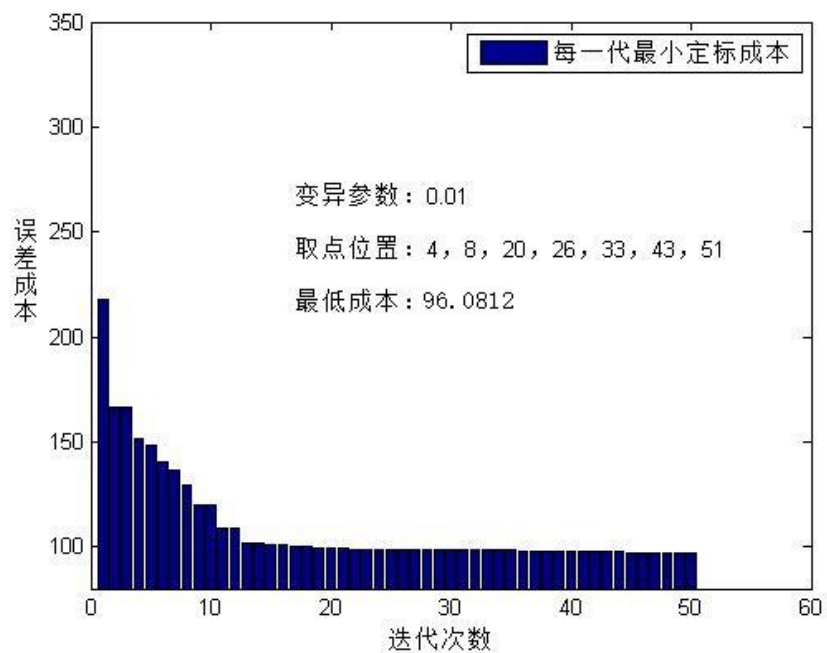
### 4.2 探究变异概率参数对结果的影响：

变异概率也是影响新个体产生的一个因素，变异概率小，产生新个体少；变异概率太大，又会使遗传算法变成随机搜索。

由上图可知，当遗传代数为 30 代后结果趋于稳定，因而取种群大小仍为 150，遗传代数为 50 代，遗传概率分别为 0.01, 0.001，进行比较：



图六 探究变异概率参数对结果的影响 (0.01)



图七探究变异概率参数对结果的影响 (0.001)

分析:

当变异概率为 0.001 时, 所得最低成本为 106.1395, 此时遗传代数为 23,

当变异概率为 0.01 时, 所得最低成本为 96.0812, 此时遗传代数为 26,

与所查资料结果相符, 变异概率偏大或偏小, 所得最低成本均高于变异概率为 0.05 时的最低成本 (95.2147)。

当变异概率为 0.1 与 0.01 时所得最低成本的遗传代数相近, 但最低成本相差较大。分析原因, 当运行时间相近时, 变异概率较大, 程序运行结果将较快趋向最优解。

## 5 结论

- 1) 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。经比较,当运用三次样条插值拟合法、遗传算法时,得到最低成本为 95.2147,取 7 个点,最优取点方案为 2,9,20,27,34,44,50。
- 2) 通过运行 test\_ur\_answer.m 程序计算最优解的成本,与所得最低成本一致。
- 3) 在统计推断在模数、数模转换系统中的应用实例的分析中,利用遗传算法和三次样条插值拟合法,求得某输入输出系统的特征点分布,从而得到了一种在工程应用中能够减少测量工作量,合理、高效评估产品性能的方法,这在工程应用中可以减少不必要的测量,节约时间和生产成本。

## 6 参考文献

- [1] 陈怀琛——MATLAB 及其在理工课程中的应用指南. 西安电子科技大学出版社. 2004
- [2] 冯卫国, 孙祝岭——概率论与数理统计. 上海交通大学出版社第 2 版. 2003
- [3] 上海交通大学电子工程系. 统计推断在数模、模数转换系统中的应用课程讲义
- [4] 雷英杰, 张善文——MATLAB 遗传算法工具箱及应用西安电子科技大学出版社第 2 版. 2014
- [5] 百度百科——遗传算法  
[http://baike.baidu.com/link?url=HWLx7PTTdhd5kWMsxPkyPWFDfzYduNYFEo6\\_Mi09IXNHeL12RUAok5h8wKoN-Zuvpqih8YJWxkN4giWgAONMuK](http://baike.baidu.com/link?url=HWLx7PTTdhd5kWMsxPkyPWFDfzYduNYFEo6_Mi09IXNHeL12RUAok5h8wKoN-Zuvpqih8YJWxkN4giWgAONMuK)



## 附录：程序的具体实现

### *main.m*

```
1  clear all;clc;

2  data = dlmread('20150915dataform.csv'); %读入数据

3  M = 800; N = 51;

4  samplenum = M/2;npoint = N;

5  global x;

6  x = zeros(1,npoint);

7  global originY;

8  global tempMaxChosenY; %成本最小时的取点方案

9  global tempMax; %每一代的最小成本

10 global average;

11 originY = zeros(samplenum,npoint);

12 calculateY = zeros(400,51); %记录拟合 y 值

13 S = 150; %种群数量

14 G = 500; %遗传代数

15 Result = zeros(1,S);

16 for i = 1:samplenum

17     x(1,:) = data(i*2-1,:);

18     originY(i,:) = data(2*i,:);

19 end

20 RecordtempMax = [];

21 Recordaverage = [];

22 Gene = round(1*rand(S,51)); %种群初始化

23 for generation = 1:G %遗传算法

24     Result = fitness(S,Result,Gene); %成本计算

25     son = selection(S,Result,Gene); %淘汰选择

26     sonCross = crossover(son,S); %交叉

27     sonMutate = mutate(sonCross,S); %变异
```

```

28     Gene = sonMutate;                %子代基因
29     generation
30     tempMaxChosenY
31     tempMax
32     RecordtempMax = [RecordtempMax,tempMax];%成本值
33     Recordaverage = [Recordaverage,average];%平均值
34     average
35 end
36 bar(RecordtempMax)%画出各代成本变化趋势的条形图
37 axis([0 60 80 350]);%建立坐标轴

```

### *fitness.m*

```

1  function Result = fitness(M,Result,Gene)
2  global originY;
3  global x;
4  global tempMax;
5  tempMax = 12*51+400*51*25;
6  global tempMaxChosenY;
7  global average;
8  average = 0;
9  for m=1:M
10     [Xtemp,Ytemp] = find(Gene(m,:)'~=0);
11     count = length(Xtemp);
12     tempChosenX = Ytemp.*0.1+4.9;
13     ChosenY = originY(:,Ytemp);
14     calculateY = spline(tempChosenX,ChosenY,x);    %三次样条拟合计算
15     errorCount = abs(calculateY-originY);          %误差成本
16     le0_4=(errorCount<=0.4);
17     le0_6=(errorCount<=0.6);

```

```

18     le0_8=(errorCount<=0.8);
19     le1_0=(errorCount<=1);
20     le2_0=(errorCount<=2);
21     le3_0=(errorCount<=3);
22     le5_0=(errorCount<=5);
23     g5_0=(errorCount>5);
24     %将不同情况下的成本计算并求和
25     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;
26     si = sum(sij,2);
27     Result(1,m) = sum(si)/400+12*count; %平均成本
28     %比较得出该代最小成本
29     if Result(1,m)<tempMax
30         tempMaxChosenx = tempChosenX;
31         tempMaxChosenY = Ytemp;
32         tempMax = Result(1,m);
33     end
34 end
35 average = sum(Result)/M;
36 out = Result;
37 end

```

### *selection.m*

```

1     function son = selection(num,m,parent)
2     m = m.^2;
3     sum1 = sum(m);
4     m = 1./m;
5     m = sum1*m;
6     sum2 = sum(m);

```

```

7   m = 1/sum2*m;
8   adp = zeros(1,num)+1;
9   son = zeros(num,51); %过渡种群
10  b = m(1);
11  e = 1; %记录适应度最小的个体的位置
12  for i = 2:1:num
13      if (m(i)>b) %寻找适应度最大的个体
14          e = i;
15          b = m(i);
16      end
17      m(i) = m(i-1)+m(i);
18  end
19  for i=2:1:num %保留第一个最优个体
20      for j = 1:1:num
21          c = rand;
22          if (c<m(j) && j==1)
23              adp(i)=j;
24              continue;
25          end
26          if (c<m(j) && c>=m(j-1))
27              adp(i) = j;
28              continue;
29          end
30      end
31  end
32  son(1,:) = parent(e,:);
33  for i =2:1:num
34      son(i,:) = parent(adp(i),:);
35  end
36  out = son;

```

37 end

### *crossover.m*

```
1 function m = crossover(m,num)
2 k = num/2;
3 pos = randi(51,k,1);
4 for i = 1:1:k
5     if (pos(i)==51)
6         continue
7     end
8     %对两个基因进行交叉处理
9     tmp1 = [m(i+1,1:pos(i)),m(num-i+1,(pos(i)+1):51)];
10    tmp2 = [m(num-i+1,1:pos(i)),m(i+1,(pos(i)+1):51)];
11    if(i~=1)
12        m(i,:) = tmp1;
13    end
14    m(num-i+1,:) = tmp2;
15 end
16 out = m;
17 end
```

### *mutate.m*

```
1 function m2 = mutate(m2,num)
2 for i = 1:51
3     for j = 2:num    %每个种群保留最优个体
4         if (rand<0.05)    %变异概率
5             m2(j,i) = 1-m2(j,i);    %0, 1 变异
6         end
7     end
```

```
8   end
9   out = m2;    %m2 为最终种群
10  end
```

### *test\_ur\_answer.m*

```
1   my_answer=[ 1,11,21,31,41,51 ];%把你的选点组合填写在此
2   my_answer_n=size(my_answer,2);
3   % 标准样本原始数据读入
4   minput=dlmread('20150915dataform.csv');
5   [M,N]=size(minput);
6   nsample=M/2; npoint=N;
7   x=zeros(nsample,npoint);
8   y0=zeros(nsample,npoint);
9   y1=zeros(nsample,npoint);
10  for i=1:nsample
11  x(i,:)=minput(2*i-1,:);
12  y0(i,:)=minput(2*i,:);
13  end
14  my_answer_gene=zeros(1,npoint);
15  my_answer_gene(my_answer)=1;
16  % 定标计算
17  index_temp=logical(my_answer_gene);
18  x_optimal=x(:,index_temp);
19  y0_optimal=y0(:,index_temp);
20  for j=1:nsample
21  y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
22  end
23  % 成本计算
24  Q=12;
```

```

25  errabs=abs(y0-y1);
26  le0_4=(errabs<=0.4);
27  le0_6=(errabs<=0.6);
28  le0_8=(errabs<=0.8);
29  le1_0=(errabs<=1);
30  le2_0=(errabs<=2);
31  le3_0=(errabs<=3);
32  le5_0=(errabs<=5);
33  g5_0=(errabs>5);
34  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;
35  si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
36  cost=sum(si)/nsample;
37  % 显示结果
38  fprintf('\n 经计算，你的答案对应的总体成本为%.2f\n',cost);

```

### *mycurvefitting.m*

```

1  function y1 = mycurvefitting( x_premea,y0_premea )
2  x=[5.0:0.1:10.0];
3  y1=interp1(x_premea,y0_premea,x,'spline');
4  end

```