
统计推断在数模转换系统中的应用

题目：统计推断在数模转换系统中的应用

作者：张超 5130309366

周梦蝶 5130309134

组别：02 组

学院：电子信息与电气工程

日期：2014 年 12 月 10 号

摘要： 本文是上海交通大学电子信息与电气工程学院课程设计《统计推断在数模、模数转换系统中的应用》的课程论文。对于某种装置系统，输入和输出存在一定的受控关系，但是这种受控关系是非线性的并且个体差异较大。因此在大批量生产中，需要考虑到该受控关系。在工程实际中，若对每个样品都进行数值的完整测定，则既费时又消耗成本。若能减少需要观测的数值组数量，则可以提高工效。在限测 7 点事前观测值的情况下，我们需要研究能否获得得分不低于 94 分的估测方法，并提出合理方案与实践方法。

关键词： 统计推断，遗传算法，三次样条插值，多项式拟合

Application of Statistical Inference in DA Inverting System

ABSTRACT:

This article is for Course Design-Application of Statistical Inference in AD&DA Inverting System, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. For a unit system, there are certain controlled relationship between input and output, but this kind of controlled relationship is nonlinear and individual differences. So in mass production, the need to consider the relationship between the controlled. In engineering practice, if the value of each sample complete measurement, is both time consuming and cost. If can reduce the number of numerical group needs observation, can improve the work efficiency. Of the prior observations at 7 in the limit of test cases, we need to study whether we can get the score of no less than 94 points of estimation methods, and put forward reasonable scheme and practical method.

Key words: statistical inference, genetic algorithms, cubic spline interpolation, the polynomial fitting

1 引言

本次试验中，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 研究装置以及已得数据介绍

这种电子装置有 51 个有效控制点，对应输出电压 $U = 5.0V \ 5.1V \ \cdots \ 10.0V$ ，也就是说在使用时不会被要求输出类似 5.05V、7.87V 的电压。每个样品的实验数据包含 51 组数值，对应 $U = 5.0V \ 5.1V \ \cdots \ 10.0V$ 。我们实际制作了 469 个此种装置样品，并通过实验测量，获得了每个样品的特性数据。

1.2 任务概述

在限测 7 点事前观测值的情况下，研究能否获得总成本最低的估测方法，并提出合理方案与实践方法。

2 对解决问题的几种算法的讨论

如何选取特征点，同时选择什么样的方式找出曲线才能最好的反映数据的分布是我们需要讨论的两个问题。

2. 1事前观测点数据处理方法的选取

2. 1. 1 三段多项式拟合

此方法为拟合的常用简单方法，即设一在区间 $[a, b]$ 上的 n 阶多项式函数

$$P(x) = \sum_{k=0}^n a_k x^k$$

其中 $a=x_0 < x_1 < \dots < x_n = b$ 是数据点，则它为区间 $[a, b]$ 上对数据点拟合的多项式。由定义可看出，它有 $n+1$ 个待定系数。确定这些系数 a_k 使

$$D = \sum_{k=1}^{53} [y_k - P(x_k)]^2$$

最小。这时就得到误差平方拟合多项式。可以把它认为是推断函数。同样，我们利用matlab中已有的函数

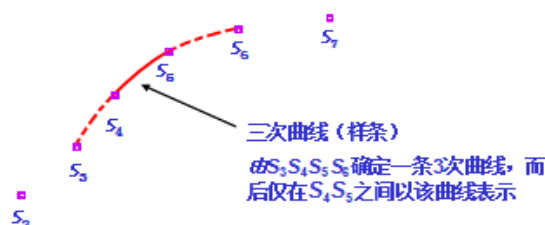
$$a = \text{polyfit}(x_0, y_0, m)$$

进行多项式拟合。其中输入参数 x_0, y_0 为要拟合的数据， m 为拟合多项式的次数，输出参数 a 为拟合多项式系数。多项式在 x 处的值 y 可用matlab中的 $y = \text{polyval}(a, x)$ 函数进行计算。

多项式拟合的次数在一定范围内越高，方差等参数越小，拟合曲线与实际测量点的相关性越高，拟合程度越好。但次数的增高导致算法运行时间的延长，效率的降低，而且当次数超过一定范围时，甚至适得其反。例如用5, 6次函数拟合，运行时间是很难让人接受的，所得结果误差反而越来越大，可见不一定次数越高，拟合曲线就越接近实际曲线。反复测试后，3次曲线拟合的效果最好。

2. 1. 2 三次样条插值法

三次样条插值拟合，即用6段三次曲线来近似表示。选出7个特征点，对于中间的点而言，取中间点加上两侧的与中间点相邻的两个点作为研究对象。通过四个点确定出函数关系方程作为中间点之间曲线方程，以此类推。例如，图五中对于 S_3, S_6 之间而言，由 S_3, S_4, S_5, S_6 确定一条3次曲线，而后仅在 S_4, S_5 之间以该曲线表示，其它点同理。而对于边缘上的点而言，如对点 S_1 和 S_2 ，由 S_1, S_2, S_3, S_4 确定一条3次曲线，而后仅在 S_1, S_2 之间以该曲线表示。其优点在于它对于相邻两点之间都采用一条三次曲线来表示，这样就能使整体曲线非常平滑，且准确度高。但是该方案的缺点在于每两个点之间都用一条三次曲线拟合，其过程非常复杂，对于程序的编写和matlab的运算量都是一大考验。



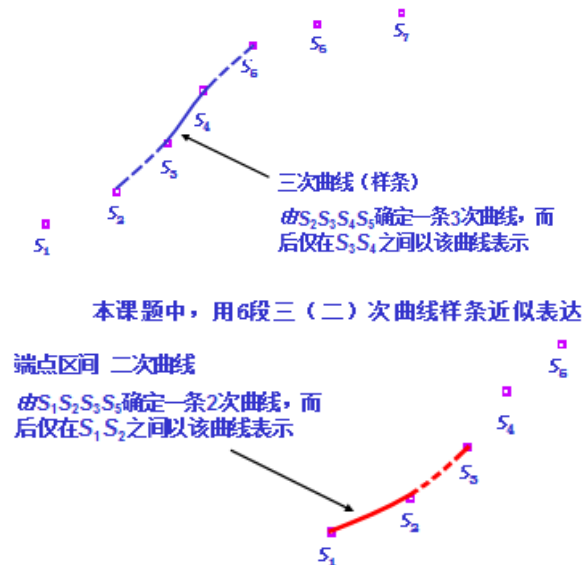


图 1 三次样条插值法示意图^[1]

2.2 选取 7 个特征点

2.2.1 暴力穷举法（不可取）

总共需要尝试（51 中取 7 的组合）有 1 亿多种不同组合，求解十分复杂。因此不可行。为了减少数据量，我们可以通过缩小范围的方法找到拟合点的大致位置，只在拟合点附近进行嵌套循环，最终找到最优解。

2.2.2 不等间隔取点

在一组数据中选取七个点拟合的问题，一般的想法就是 $C_{51}^7=154143080$ 把每七个点都遍历一边，对于普通的计算机来说很难完成。因此，我们小组先每隔一个间断选取一个范围，一共选取七个范围进行枚举运算。

2.2.3 模拟退火算法^[2]（SA）

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。根据 Metropolis 准则，粒子在温度 T 时趋于平衡的概率为 $e(-\Delta E/(kT))$ ，其中 E 为温度 T 时的内能， ΔE 为其改变量， k 为 Boltzmann 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，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表(Cooling Schedule)控制，包括控制参数的初值 t 及其衰减因子 Δt 、每个 t 值时的迭代次数 L 和停止条件。

2.2.3 遗传算法（genetic algorithm）

（1）遗传算法的简介

遗传算法是从代表问题可能潜在的解集的一个种群（population）开始的，而一个种群则由经过基因（gene）编码的一定数目的个体(individual)组成。每个个体实际上是染色体(chromosome)带有特征的实体。染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（即基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此，在一开始需要实现从表现型到基因型的映射即编码工作。由于仿照基因编码的工作很复杂，我们往往进行简化，如二进制编码，初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代（generation）演化产生出越来越

越好的近似解，在每一代，根据问题域中个体的适应度（fitness）大小选择（selection）个体，并借助于自然遗传学的遗传算子（genetic operators）进行组合交叉（crossover）和变异（mutation），产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后代种群比前代更加适应于环境，末代种群中的最优个体经过解码（decoding），可以作为问题近似最优解。

遗传算法是具有“生成+检测”的迭代过程的搜索算法。基本流程如图 3 所示。可见遗传算法是一种群体型操作，该操作以群体中所有个体为对象。选择（selection）、交叉（crossover）和变异（mutation）是遗传算法的三个主要操作算子。

遗传算法是模仿遗传二字进行运算的。借阅生物界的遗传规律形成的算法。优胜略汰，适者生存，是一种淘汰型的搜索和进化，生物界的“程序”编译运行着，同时知道下一步该怎样运行，对哪些单位进行何种的筛选，拥有启发式，自动获取优化的方向和空间。即为程序完善自身运行的过程^[3]。

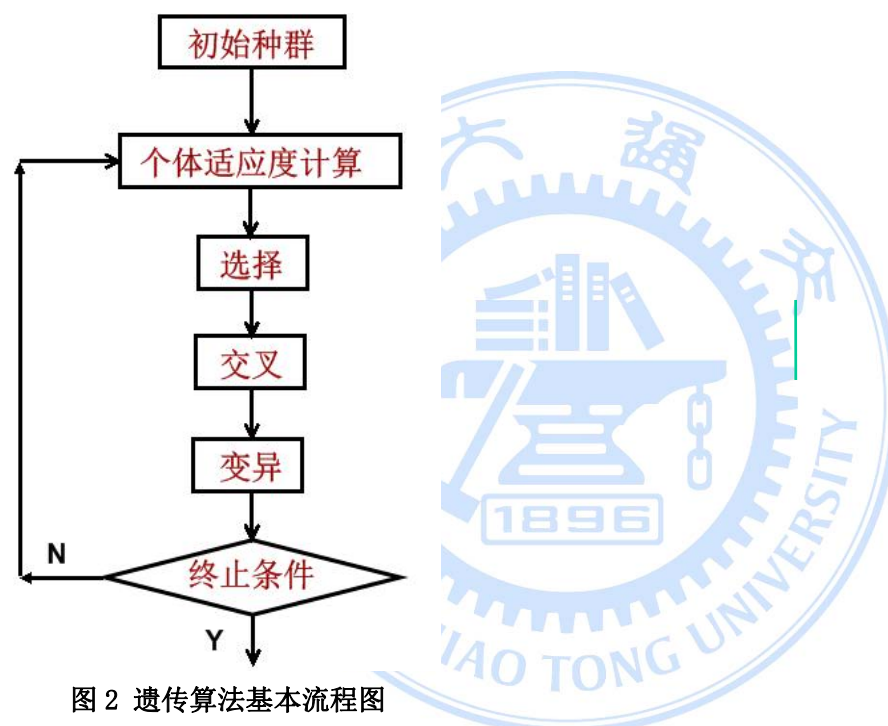


图 2 遗传算法基本流程图

创建初始种群(需要筛选的群体)，个体适应度计算(计算 94 分以上的函数)，选择(code)，交叉（程序自动优化过程，一点或则多点交叉），变异（配合交叉的过程，减弱局部解的弊病），选择（决定那些数据可以进行“下一代”）

（2）遗传算法的具体应用

1. 主要操作

（1）（1）创建初始种群，种群规模取 100。随机选择得到一组解（基因），每一个基因用一组一维数组表示，如编码方式 1 { p1 , p2, p3, p4, p5, p6, p7}，即采用十进制数组编码法。

（2）个体适应度的计算：对于所选择的 100 组基因评估适应度

（3）繁殖：通过基因适应度的大小所占比例随机产生选择结果，然后配对交叉，基因在第四个和第五个数据点处断开。产生后代，之后有随机进行基因突变，由此得到下一代种群。

在产生下一代时会产生“非法后代”，如：

{ 1, 8, 23, 35, 43, 47, 52 }	{ 1, 8, 23, 35, 31, 35, 51 }
{ 2, 6, 17, 28, 31, 35, 51 }	{ 2, 6, 17, 28, 43, 47, 52 }

因此，在遇到这种情况时，要采取合适的处理方法。可以将其中一个重复的随即换成一个不会造成非法的特征点。

同时，产生下一代时也要有随机的基因突变。即某一个基因中的某个数突变为另外的数值。

注意：交叉后会产生“非法后代”，要有合适处理规则

编码方式 2: { g1 , g2, g3, ..., g52, g53}, 二进制数组编码法。个体构成总群必须经过解码，已转换为原问题的空间的决策变量构成的种群。方能计算相应的适应度。

合法个体含 7 位 1 { 0, 1, 0, 0, 0, 1, ..., 0, 1, 0 }

交叉后也会有“非法后代”现象，要有合适处理规则

(4) 选择下一代，是否有成功解集的出现。出现则问题解决，否则就自动重新运行过程。

2. 编码

流程图

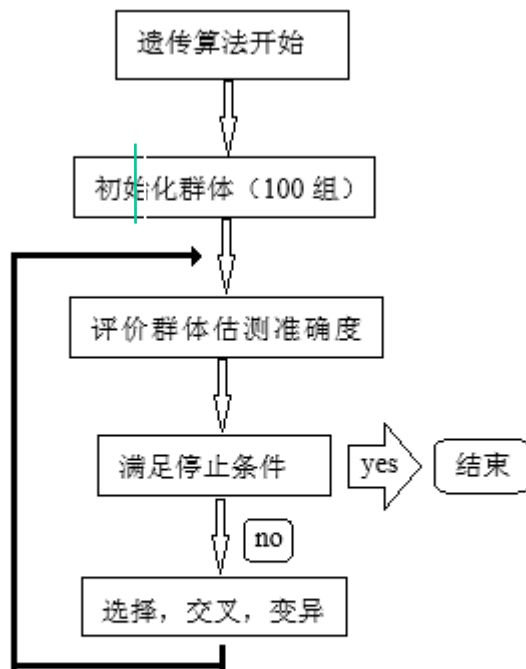


图 3 遗传算法流程图

3 课题模型的评判模型

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

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

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。

- 校准方案总体成本

按式（3）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。

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

总体成本较低的校准方案，认定为较优方案。

4 数据结果与分析

4.1 数据结果

1. 因为暴力穷举法耗费时间太长，因此我组在稍微尝试之后，仅能列出代码，而无最终结果。

2. 不等间隔取点法：利用三次样条插值法，得到最低误差组为 [2 8 16 25 33 43 50]，经计算得出最低成本为 95.4979。

3. 不等间隔取点法：利用多项式拟合的方法，得到最低误差组为 [2 6 16 25 34 43 49]，经计算得出最低成本为 133.2143

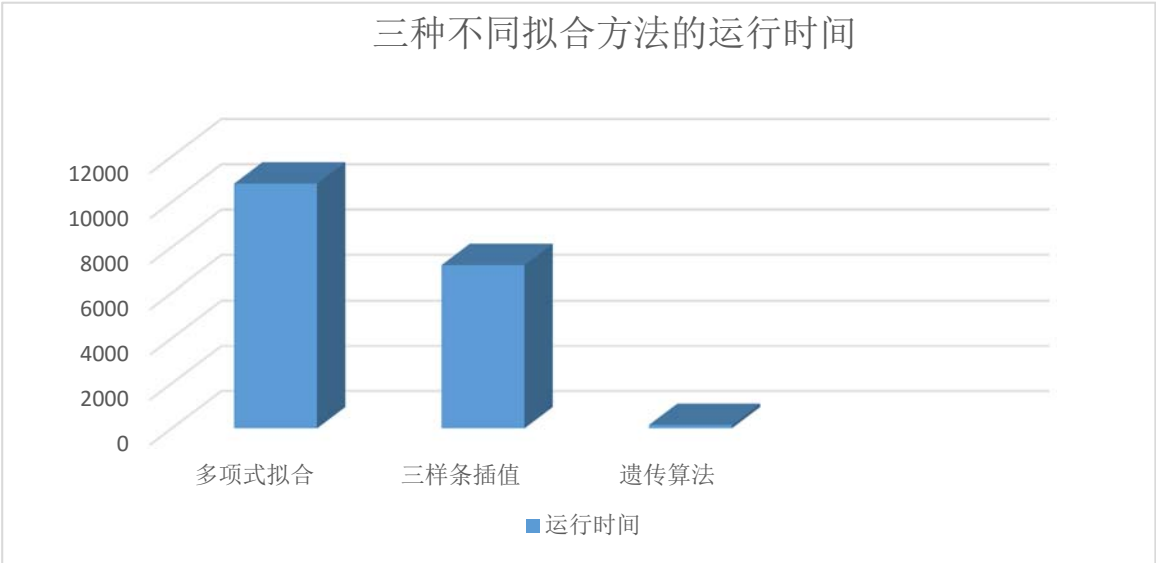
4. 遗传算法：利用 GA，得到的最低误差组为 [2 9 21 31 41 51]，经计算得出的最低成本为 89.7100

4.2 运算结果分析与讨论

我们小组对如何在较短时间内选择所用时间最短，得分最高的方法进行了探究。我们通过查阅资料得知三次样条插值法在对数据点的拟合上更优于三次多项式拟合法，我们在

拟合方式上分别选用了多项式拟合和样条插值的方法，数据表明，三样条插值法确是优于多项式拟合。然后通过遗传算法来编写程序，对比其与不等间隔取点法的结果的优劣性，从而确立最佳特征点选取方案。图4反映了三种方法的运行时间对比。

从运算速度我们可以看到，遗传算法速度具有压倒性的优越性。



在本文中，我们在进行遗传算法的运行中，记录的是6点取样，为了寻找到最佳的取店方案，我们小组还进行了其他5点和7点的取点方式。其中点数取得越少运行速度越快，但是其成本运算越高，综合考虑下，我们最终选择了6点的取点方式，因为在三种取点方式之中，6点的运行速度较7点取样更有优势，与此同时，其成本也与7点取样基本将近，模拟次数多之后，在该数据量的情况下，相差很小。同时点的成本明显比5点低。综合考虑下，在该数据样本之下，还是选择了6点取点的方式。图5显示了5、6、7三种取点数的拟合成本与运行时间的对比。

图 4

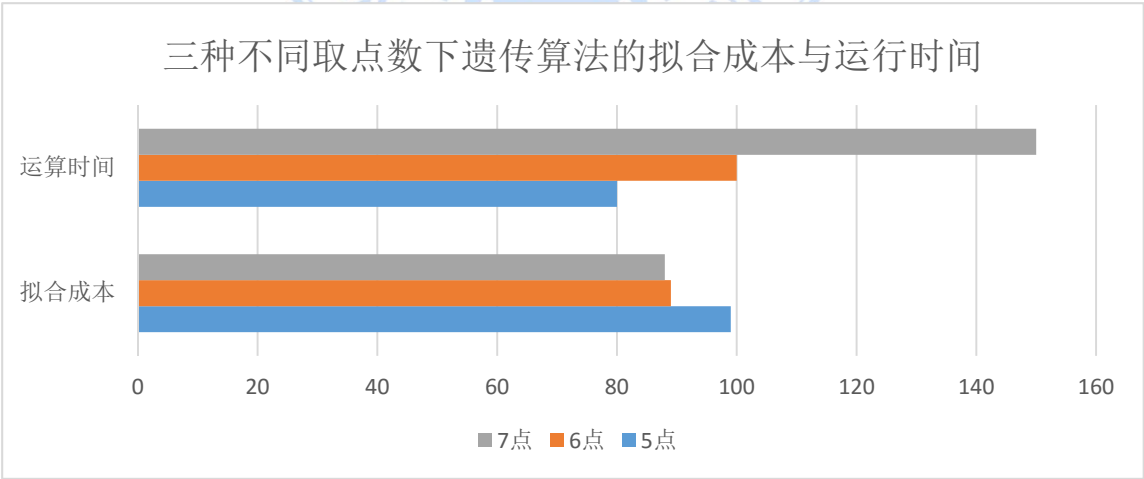


图 5

5 课题结论

通过不等间隔取点和遗传算法两种不同的算法，其中不等间隔取点方式包括多项式拟合以及三样条插值法。遗传算法包括5、6、7个事前观测点的不同取点数方式。通过计算成

本大小和程序运行时间，比较了不等间隔取点拟合和遗传算法不同取点数的结果，结果显示遗传算法在准确度与运行时间上更具优势，并且在该数据样本的前提之下，6点取样更为合适。

因此我们得出了“在该问题中采用遗传算法更为合理，且用6点取点的方式更加合理”这一结论。

6 结束语

通过对统计推断在模数、数模转换系统中的应用实例的分析，我们得到了一种在工程应用中能够减少测量工作量并快速获得类似于本例中 $D-U$ 函数的一般方法。即通过遗传算法和三次样条插值拟合法，求得某个输入输出系统的特征点分布。对于之后的同样系统特性的测量，我们只需要测量器在特征点的输入输出特性，即可获得该系统的输入输出关系。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。

7 参考文献

- [1] 百度文库《第九章插值与拟合》[J/OL]
- [2] 飞思科技产品研发中心 MATLAB7基础与提高 电子工业出版社 2007-5
- [3] 袁焱 统计推断课程讲座 上海交通大学 2013-9至2013-10

8 致谢

感谢袁焱老师的讲座指导。
感谢为课程过程中互相帮助的同学。
感谢学校安排这门课程。



附录:

1. 暴力穷举法代码:

```
warning off all;
clear;
clc;
tic;
alldata = xlsread('data.xls','Sheet1','A1:AY938','basic');
Min = 1026;
for p1 = 1:45
    for p2 = (p1+1):46
        for p3 = (p2+1):47
            for p4 = (p3+1):48
                for p5 = (p4+1):49
                    for p6 = (p5+1):50
                        for p7 = (p5+1):51
                            Points = [p1 p2 p3 p4 p5 p6 p7];
                            Scores1=curvefits1(alldata,Points)
                            if Scores1< Min
                                Min = Scores1;
                                disp(Points);
                                disp(Min);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
toc;
```

2. 不等间隔取点法代码:

```
warning off all;
clear;
clc;
tic;
alldata = xlsread('data.xls','Sheet1','A1:AY938','basic');
Min = 1030;
for p1 = 2:4
    for p2 = 5:9
        for p3 = 12:16
            for p4 = 21:25
                for p5 = 31:37
                    for p6 = 43:46
```

```
    disp(Points);  
end;  
toc;
```

```
function [pop k] =creation(popsizel)  
k = 0;  
pop = ones(popsizel,7)*100;  
for i = 1:popsizel  
    pop(i,1) = randi([1,5]);  
    pop(i,2) = randi([6,10]);  
    pop(i,3) = randi([11,18]);  
    pop(i,4) = randi([19,30]);  
    pop(i,5) = randi([31,40]);  
    pop(i,6) = randi([41,50]);
```

```
end;  
for t = 1:popsizel-1  
    for m = (t+1) : popsizel  
        for n = 1:6  
            if (pop(t,n)~= pop(m,n))  
                k = 1;  
                break;  
            end;  
        end;  
    end;  
end;  
end;
```

```
end
```

```
function [ new_pop] =mutationpop(pop ,popsizel, p_mutation,alldata)  
new_pop = pop;  
Array = ones(1,51) + randperm(51);  
for i = 1: popsizel  
    if(rand < p_mutation)  
        point_c = randi([1,6]);  
        remain = setdiff(Array,pop(i,1:6));  
        new_pop(i,point_c) = remain(randi([1,44]));  
    end;  
end;
```

```

new_pop = sort(new_pop,2);
for p = 1:popsize
    new_pop(p,7) = curvefits(alldata,new_pop(p,1:6));
end;

```

```

end

```

```

function [ new_pop ] = cross(pop,alldata,p_cross)
[px py] = size(pop);
new_pop = pop;
for i=1:2:(px-1)
    if(rand<p_cross)
        C_point = randi([2,6]);
        score1 = pop(i,7)+pop(i+1,7);
        new_pop(i,1:6) = [pop(i,1:C_point-1) pop(i+1,C_point:(py-1))];
        new_pop(i+1,1:6) = [pop(i+1,1:C_point-1) pop(i,C_point:(py-
1))];
        new_pop(i,7) = curvefits(alldata,new_pop(i,1:6));
        new_pop(i+1,7) = curvefits(alldata,new_pop(i+1,1:6));
        score2 = new_pop(i,7)+new_pop(i+1,7);
        if(score2>score1)
            new_pop(i,:)=pop(i,:);
            new_pop(i+1,:)=pop(i+1,:);
        end;
    end;
end;
end;
end

```

```

function Score = curvefits(alldata,points)
Xi = zeros(1,51);
Score = 0;
for i=1:100
    PointX = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2))
alldata(2*i-1,points(3)) alldata(2*i-1,points(4)) alldata(2*i-
1,points(5)) alldata(2*i-1,points(6)) ];
    PointY = [alldata(2*i,points(1)) alldata(2*i,points(2))
alldata(2*i,points(3)) alldata(2*i,points(4)) alldata(2*i,points(5))
alldata(2*i,points(6)) ];
    for m = 1:51

```

```

        Xi(m) = alldata(2*i-1,m);
    end;
    y = interp1(PointX,PointY,Xi,'spline');
    Diff = y - alldata(2*i,:);
    score = 72;
    for t=1:51
        if abs(Diff(t))<=0.5
            score=score+0;
        elseif abs(Diff(t))<=1
            score=score+0.5;
        elseif abs(Diff(t))<=2
            score=score+1.5;
        elseif abs(Diff(t))<=3
            score=score+6;
        elseif abs(Diff(t))<=5
            score=score+12;
        else score=score+25;
        end;
    end;
    Score = Score+score;
end;
Score = Score/100;

```

4.多项式拟合功能函数:

```

function [ Scores1 ] = curvefits1(alldata,points)
Scores1 = 0;
for i=1:469
    x = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2))
alldata(2*i-1,points(3)) alldata(2*i-1,points(4)) alldata(2*i-
1,points(5)) alldata(2*i-1,points(6)) alldata(2*i-1,points(7))];
    z = [alldata(2*i,points(1)) alldata(2*i,points(2))
alldata(2*i,points(3)) alldata(2*i,points(4))
alldata(2*i,points(5)) alldata(2*i,points(6))
alldata(2*i,points(7))];
    p3 =polyfit(x,z,3);
    y = polyval(p3,alldata(2*i-1,:));

    Diff = y - alldata(2*i,:);
    score = 12*7;
    for t=1:51
        if abs(Diff(t))<=0.5
            score=score+0;

```

```

elseif abs(Diff(t))<=1
    score=score+0.5;
elseif abs(Diff(t))<=2
    score=score+1.5;
elseif abs(Diff(t))<=3
    score=score+6;
elseif abs(Diff(t))<=5
    score=score+12;
elseif abs(Diff(t))>5
    score=score+25;        end;
end;
Scores1 = Scores1 +score;
end;
Scores1 = Scores1/469;

```

5.三次样条插值拟合功能函数:

```

function Scores = curvefits(alldata,points)
Xi = ones(1,51);
Scores = 0;
for i=1:469
    PointX = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2)) alldata(2*i-1,points(3)) alldata(2*i-1,points(4)) alldata(2*i-1,points(5)) alldata(2*i-1,points(6)) alldata(2*i-1,points(7))];
    PointY = [alldata(2*i,points(1)) alldata(2*i,points(2)) alldata(2*i,points(3)) alldata(2*i,points(4)) alldata(2*i,points(5)) alldata(2*i,points(6)) alldata(2*i,points(7))];
    for m = 1:51
        Xi(m) = alldata(2*i-1,m);
    end;
    y = interp1(PointX,PointY,Xi,'spline');
    Diff = y - alldata(2*i,:);
    score = 12*7;
    for t=1:51
        if abs(Diff(t))<=0.5
            score=score+0;
        elseif abs(Diff(t))<=1
            score=score+0.5;
        elseif abs(Diff(t))<=2
            score=score+1.5;
        elseif abs(Diff(t))<=3
            score=score+6;
        elseif abs(Diff(t))<=5
            score=score+12;
        elseif abs(Diff(t))>5
            score=score+25;
        end;
    end;
end;
Scores = Scores/469;

```

```
        end;  
    end;  
    Scores = Scores + score;  
end;  
Scores = Scores/469;
```

