统计推断在数模转换系统中的应用

组号 16 徐鹏飞 5132119029 胡悦 5130309599

摘要：在本次课题中我们首先对 469 组数据进行观察，初步确定用奇数次多项式拟合的方法进行拟合，然后根据遗传算法搜索最优解。

关键词：多项式拟合，三次样条插值，遗传算法

1 引言

该课题来自于工科创 5 的项目内容：DC-DC 开关电源的控制。单片机程序中要使用 D-U 对应关系（表达式或数据表），但是每块电路板的 D-U 特性关系可能都是独一无二的，必须逐块测定，可以耐心地逐点测定 51 个电压设定对应的占空比 D 值。若该电路板投入工业大规模批量生产，也必须逐块测定，那么将耗费大量的人力物力，所以能否仅从几个点（比如 7 个点）就可以推测出总共 51 个点的 D-U 对应关系，就引出了该统计推断的研究课题。

2 数据样本的分析

本课题共提供工科创的数据 469 组，用 origin 画出多组数据的图像后，我们发现图像大致分为三个部分，1-20,35-51 区间的弯曲度比较大，20-35 区间的弯曲度比较小。

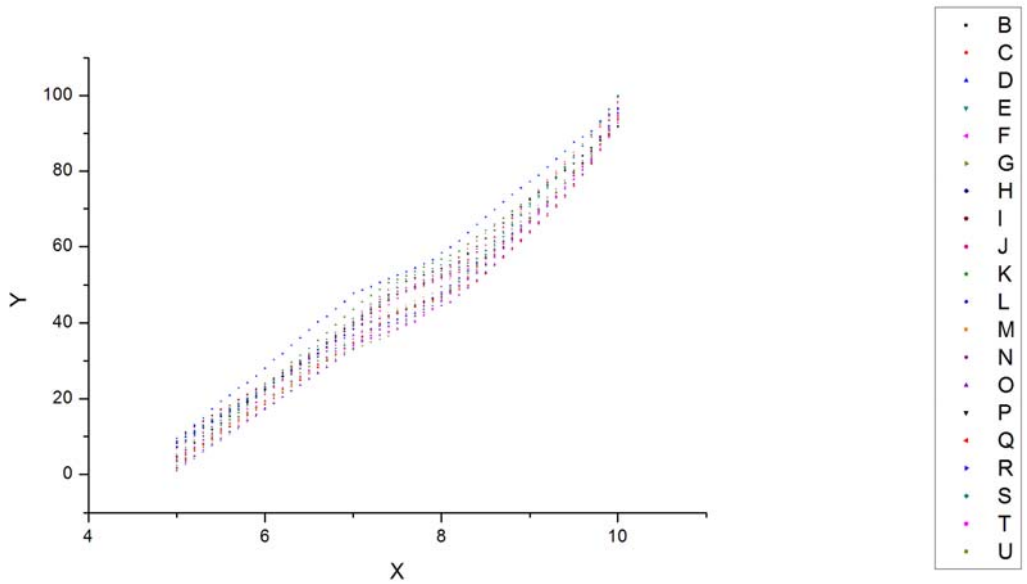


图 2-1 用 origin 得到的 X-Y 的拟合曲线

3 拟合方式选择

在本实验中，采用多项式拟合（三次、五次）和样条插值法进行拟合，在此进行初步的分析，拟合效果的评判将在算法得到具体拟合方案后在进行具体分析。

3.1 多项式拟合

拟合方式的初步选择，用所有样本点进行拟合，此时不用考虑取点成本，绘制出三次、四次、五次、六次、七次的拟合曲线，并得到相应的误差数据。由下表数据可知，拟合次数越高，误差越小，拟合精度越高。但这是在不考虑取点成本情况下的拟合方案，随着点数增加，取点数也相应增加，成本增加，而且由成本函数的选择可知取样成本的影响较大。故并非多项式次数越高，平均成本越小。

表 3-1 多项式拟合次数与单个样本残差和

拟合次数	3	4	5	6
单个样本平均残差和	42.2085	8.6707	7.3568	3.1225

3.2 样条插值

样条插值法，当使用所有样本点时，残差和为零。就此点来说优于多项式拟合。

4 特征点的选取

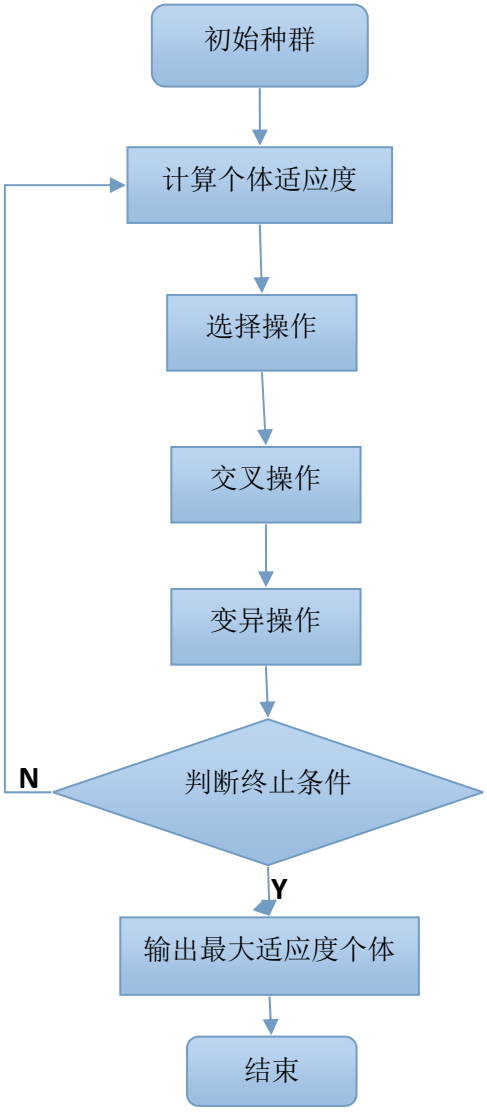
前面已经讨论过，因为测量点有 51 个，暴力穷举在实践中效率太低，需要在仅仅知道少数个点的情况下推测出整体样本的特性，用于下一步的定标检测。

为了找到合适的定标方案，我们使用了遗传算法。

4.1 遗传算法

寻找最优特征点的过程其实是一个搜索全局最优解的过程，如果使用暴力穷举法，严重浪费时间空间，这里使用遗传算法是寻找最大程度接近全局最优解。

4.1.1 传统 GA 流程图



4.1.2 算法要素

（1）设计思路

把每个取点方案用 51 位二进制数对应的一维数组表示。相应位数对应为 1 则表示取出该位置的点，1 的个数即为取点的个数（比如[1，1，0，0，0……0，0]代表[1，2]这一取点方案）。通过上面的适应度函数计算种群中每种基因型的适应度及生存概率,根据随机数模拟出存活的个体及其基因型（赌轮盘法）。之后进行交叉互换，把 51 个基因分为 2 段，两两随机配对得到新个体。之后随机取若干个位置，令这些位置的基因发生变异，得到最终的第二代个体可能的基因型。之后再重复相同的操作直至循环代数足够多，循环结束过程中生存概率最大的基因型对应的 51 位二进制数对应的取点方案是该种拟合较好的取点方案。

（2）初始种群的生成

用多个随机产生的取点方案的点的序号构成的一维数组作为初始种群中的个体，考虑到算法的效率问题，平均3个点生成生一个1而不是平均2个点生成一个1。任意带入一组取点方案得到用三次样条插值和三次多项式计算成本均小于180，说明最佳取点方案应该小于180/12=15个点，而1的数量过少不利于基因多样性，故采取平均3个点生成1个1，使1的个数约为17个。考虑到基因多样性，生成的种群一般有100到200个个体。

```
a=zeros(num,51);
for i=1:num
 for j=1:17
     h=randi(3)+3*j-3;
     a(i,h)=1;
 end
end
```

（3）成本函数

该成本函数的自变量为取点方案中的点的序号构成的一维数组，输出结果为三次多项式，和3次样条插值模拟时该取点方案下469组数据对应的平均成本。

（4）适应度函数的选取

起初选取成本函数的倒数为适应度函数，发现种群进化较慢，且退化现象经常发生，由于反比例函数的性质，前期进化较快，但后期进化较慢且整个种群容易陷入局部最优解。考虑到误差的问题，取点数小于等于 4 个基本没有应用的意义。因此最低成本必然大于12*4=48。因此选取 1/（成本函数-48）为成本函数可以加快后期的进化速度，减少陷入局部最优解的几率。

```
live(u)=1000/(t-48);
```

（5）随机的配对交叉

先随机交换种群中的个体，打乱个体的排列顺序，再对相邻的个体进行配对交叉，保证了交叉配对的随机性。

```
 for u=1:floor(num/2)
        ran1=randi(num);
        ran2=randi(num);
        a([ran1,ran2],:)=a([ran2,ran1],:);
 end
 for u=1:floor(num/4)
   b(4*u-3,22:51)=a(4*u-2,22:51);
   b(4*u-2,22:51)=a(4*u-3,22:51);
   b(4*u-1,31:51)=a(4*u,31:51);
   b(4*u,31:51)=a(4*u-1,31:51);
 end
 a=b;
```

（6）防止种群退化

为防止种群退化，人为保留每一代的最优基因型。

```
    a(1,:)=best;
```

（7）跳出局部最优解

针对陷入局部最优解的情况，定义了整形数same来纪录某一代最优解持续的代数

```
    if(best==trans(bst))
```

```
        same=same+1;
    else
        same=0;
    end
```

当最优解停留在某一值大于10代时，对此最优解采取较大概率的基因突变以及交换片段结合的方法以得到更优的个体。本算法交换了最优解随机产生的位点左右两侧的基因，考虑到运行至后期，在51个基因中0的个数为44个以上，约有6/7的位点左右两侧均为0，进行了无效交换，故每次产生7个位点，使操作一次约进行1次有效片段交换（0和1的交换）。

```
if same>=10
    for u=1:7
        ran=randi(50);
        a(1,[ran,ran+1])=a(1,[ran+1,ran]);
    end
    ran=randi(51);
    a(1,ran)=1-a(1,ran);
end
```

4.2 函数的说明

f3函数：输入取的点的序号构成的一维数组，输出469组数据的平均定标成本。（三次样条插值拟合）

nihe函数：输入取的点的序号构成的一维数组，输出469组数据的平均定标成本。(三次多项式拟合)

f5函数：输入取的点的序号构成的一维数组，输出469组数据的平均定标成本。（五次多项式拟合）

he函数：输入种群中个体数，变异率，代数，输出遗传算法得出的取点方案。

transfer函数和trans函数：51位二进制数和取点方案对应的一维数组相互转换。

4.3 MATLAB代码（见附录）

5 结果分析

采用三种拟合方案，三次多项式拟合、三次样条插值法和五次多项式拟合得到三组最优解。三次多项式拟合最优解是五个点，取点方案是[3,13,24,35,48]，最小成本111.9733；五次多项式拟合最优解是七个点，取点方案是[2,5,17,25,34,44,50],最小成本107.2953；三次样条插值拟合最优解是六个点，取点方案[3,12,22,31,43,50]，最小成本92.8774。就最后平均成本分析知三次样条插值最优。

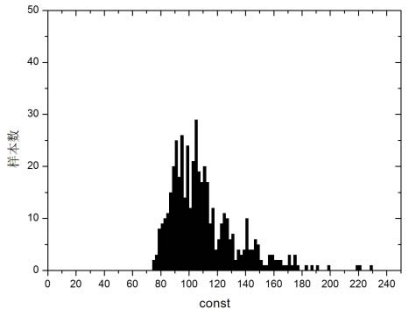


图 5-1 三次多项式-单个样本的计算成本

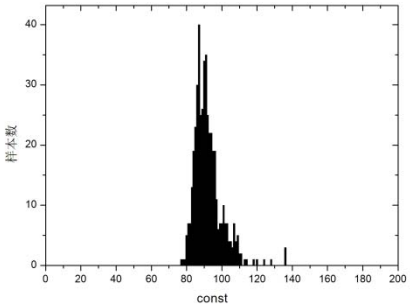


图 5-2 三次样条插值-单个样本的计算成本

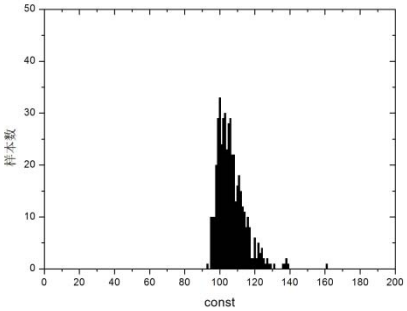


图 5-3 五次多项式-单个样本的计算成本

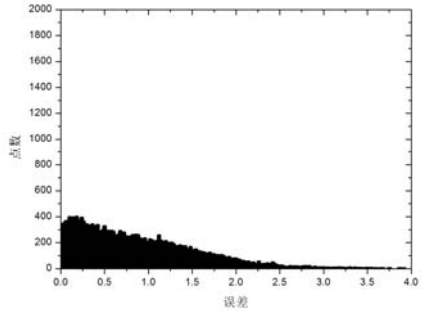


图 5-4 三次多项式-所有点的误差绝对值

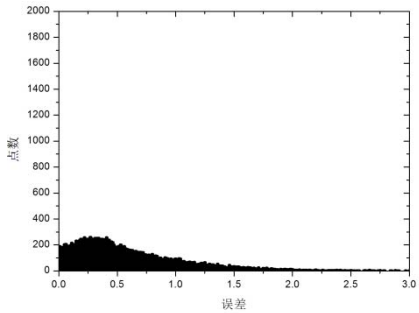


图 5-5 三次样条插值-所有点的误差绝对值

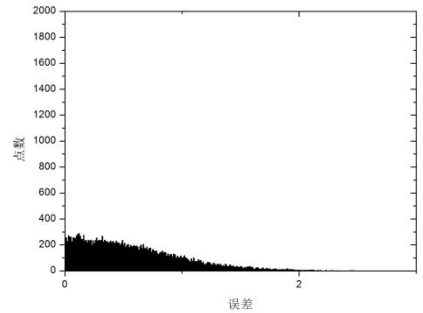


图 5-6 五次多项式-所有点的误差绝对值

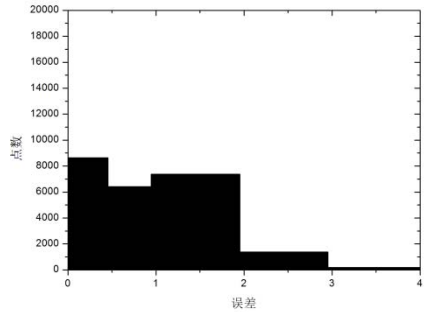


图 5-7 三次多项式-所有点的误差绝对值（按评分标准分割）

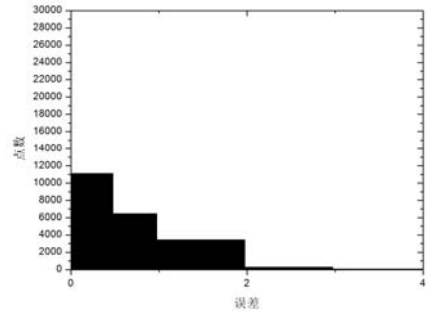


图 5-8 三次样条插值-所有点的误差绝对值（按评分标准分割）

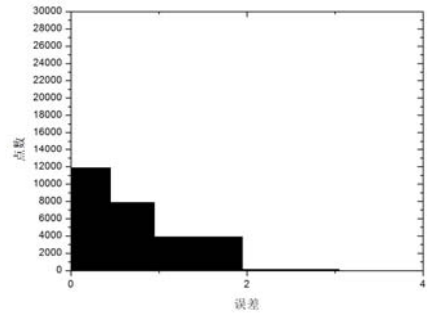


图 5-9 五次多项式-所有点的误差绝对值（按评分标准分割）

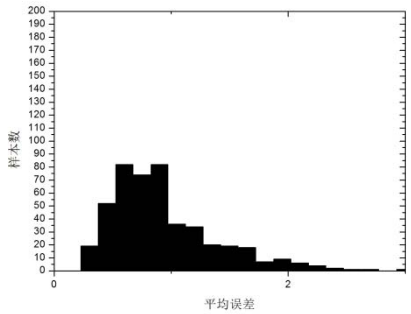


图 5-10 三次多项式-所有样本的平均误差

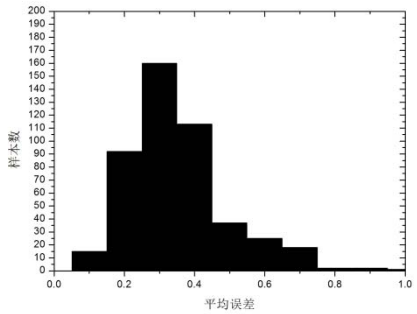


图 5-11 三次样条插值-所有点的平均误差

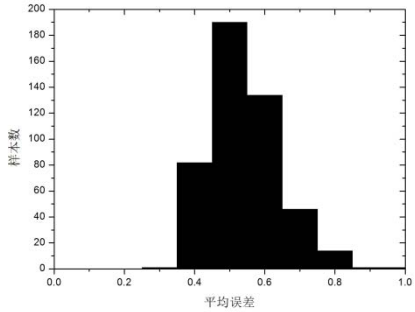


图 5-12 五次多项式-所有样本的平均误差

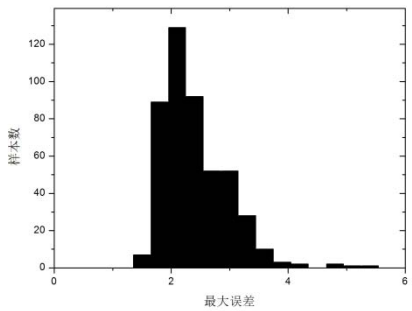
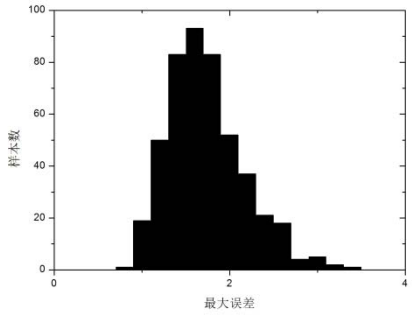


图 5-13 三次多项式-所有样本的最大误差



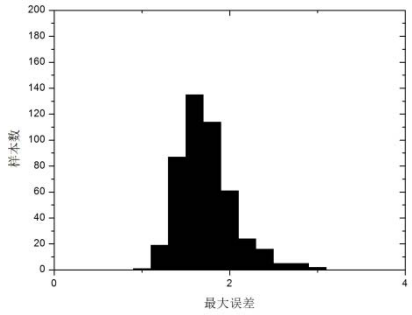


图 5-14 三次样条插值-所有样本的最大误差 **图 5-15 五次多项式-所有样本的最大误差**

图 5-1、5-2、5-3 就单个样本的计算成本而言，三次多项式拟合在 85-110 之间集中分布，三次样条插值成本集中在 85-100 左右，五次多项式拟合集中在 100-110 之间。三次样条插值比较集中，且成本较小，五次多项式拟合较前二者偏大。

图 5-4、5-5、5-6 分析所有点的误差绝对值，三次多项式拟合大部分分布在 0-2 之间，三次样条插值大部分分布在 0-1 之间，1-2 之间有少量分布，五次多项式拟合分布情况与三次样条插值类似，大部分分布在 0-1.5 之间。

图 5-7、5-8、5-9 根据评分标准的分割 0,0.5,1,2,3,5 得到三次多项式拟合误差绝对值在 0-2 之间，呈均匀分布，三次样条插值误差绝对值在 0-2 之间，呈递减分布，五次多项式拟合在 0-2 之间呈递减分布，递减趋势与三次样条插值相近。

图 5-10、5-11、5-12 分析了所有样本的平均误差，三次多项式拟合得到的误差在 0-2 之间呈峰值分布，峰值在 0.7-0.8 之间，分布相对均匀，三次样条插值拟合得到的误差在 0-0.8 之间呈单峰值分布，峰值 0.3 左右，五次多项式拟合得到的误差在 0-0.8 之间呈单峰值分布，峰值 0.5 左右。

图 5-13、5-14、5-15 分析了所有样本的最大误差，三次多项式拟合在 2 附件集中分布，三次样条插值拟合得到的样本成本在 1.75 附近集中分布，五次多项式拟合在 1.75 附近集中分布，与三次样条插值类似。、

由以上图表分析可知，三次样条插值拟合方案优于三次多项式拟合，五次多项式拟合优于三次多项式拟合。在单个样本的成本方面，五次多项式拟合由于取样成本较高，显得劣于三次拟合和三次样条插值，但在后面误差分析图示中与三次样条插值相近，优于三次多项式拟合。

6 结论

由以上分析知较优优定标方式是三次样条插值。取点方案 [3,12,22,31,43,50]，最小成本 92.8774。

附录

1 f3 函数

function z=f3(n) %n 为定标所测的点的序号构成的一维数组，输出为以样条插值函数为拟合函数的最后的平均成本%

```
M=csvread('20141010dataform.csv');
C=0;
x=zeros(469,51);
y=zeros(469,51);
f=zeros(469,51);  %拟合计算得到的 y 值
N=length(n);
for j=1:469
  x(j,:)=M(2*j-1,:);
  y(j,:)=M(2*j,:);
end
  X=x(:,n);
  Y=y(:,n);
for j=1:469
  f(j,:)=interp1(X(j,:),Y(j,:),x(j,:),'spline');
end
  err=abs(f-y);
  a1=(err<=1 & err>0.5);
  a2=(err<=2 & err>1);
  a3=(err<=3 & err>2);
  a4=(err<=5 & err>3);
  a5=(err>5);
  C=a1*0.5+a2*1.5+a3*6+a4*12+a5*25;
  z=sum(sum(C,2))/469+12*N;
```

2 nihe 函数

function z=nihe(n) %n 为定标所测的点的序号构成的一维数组，输出为以三次多项式函数为拟合函数的最后的平均成本%

```
M=csvread('20141010dataform.csv');
C=0;
x=zeros(469,51);
y=zeros(469,51);
```

```
p=zeros(469,4);

f=zeros(469,51);  %拟合计算得到的 y 值

N=length(n);
for j=1:469
  x(j,:)=M(2*j-1,:);
  y(j,:)=M(2*j,:);
end
  X=x(:,n);
  Y=y(:,n);
for j=1:469
  p(j,:)=polyfit(X(j,:),Y(j,:),3);
 for k=1:51
  f(j,k)=p(j,1)*(x(j,k))^3+p(j,2)*(x(j,k))^2+p(j,3)*(x(j,k))+p(j,4);
end
end
  err=abs(f-y);
  a1=(err<=1 & err>0.5);
  a2=(err<=2 & err>1);
  a3=(err<=3 & err>2);
  a4=(err<=5 & err>3);
  a5=(err>5);
  C=a1*0.5+a2*1.5+a3*6+a4*12+a5*25;
  z=sum(sum(C,2))/469+12*N;
```

3 f5 函数

```
function z=f5(n)   %n 为定标所测的点的序号构成的一维数组，输出为以五次多项式函数

为拟合函数的最后的平均成本%

M=csvread('20141010dataform.csv');
C=0;
x=zeros(469,51);
y=zeros(469,51);
p=zeros(469,6);

f=zeros(469,51);  %拟合计算得到的 y 值

N=length(n);
for j=1:469
  x(j,:)=M(2*j-1,:);
  y(j,:)=M(2*j,:);
end
  X=x(:,n);
  Y=y(:,n);
```

```
for j=1:469
  p(j,:)=polyfit(X(j,:),Y(j,:),5);
 for k=1:51

f(j,k)=p(j,1)*(x(j,k))^5+p(j,2)*(x(j,k))^4+p(j,3)*(x(j,k))^3+p(j,4)*(
x(j,k))^2+p(j,5)*(x(j,k))+p(j,6);
end
end
  err=abs(f-y);
a1=(err<=1 & err>0.5);
     a2=(err<=2 & err>1);
     a3=(err<=3 & err>2);
     a4=(err<=5 & err>3);
     a5=(err>5);
     C=a1*0.5+a2*1.5+a3*6+a4*12+a5*25;
     z=sum(sum(C,2))/469+12*N;
```

4 trans 函数

```
function e=trans(o) %将取点的数组转化为 51 个 1，0 组成的数组

e=zeros(1,51);
e(o)=1;
```

5 transfer 函数

```
function q=transfer(t) %把 51 个 1 或 0 构成的数组转化为取点序号构成的数组

j=0;
q=zeros(1,1);
for i=1:51
if t(i)==1
   j=j+1;
   q(j)=i;
end
end
```

6 he 函数（遗传算法主函数）

```
function he(num,gen,rate) %遗传算法,输出为最佳取点方案

%num 为种群数量，gen 为运行代数，rate 为变异率

%定义种群中的 num 个初始个体，二维数组 a 的每一行代表一个个体

%下面的为三次样条插值拟合，将下面的 f3 替换为 nihe 后即为 3 次多项式拟合
```

```
tic;
a=zeros(num,51);
b=zeros(num,51);
live=zeros(1,num);
plive=zeros(1,num);
for i=1:num
 for j=1:17
    h=randi(3)+3*j-3;
    a(i,h)=1;
 end
end
time=0;
best=a(1,:);
last=best;
bst=transfer(best);
same=0;
```

%多代繁衍可以得到较优的基因型

```
while time<gen
time=time+1;
total=0;
average=0;
```

%适应度函数，计算适应度和生存概率，得到最优的个体

```
 for u=1:num
 cur=transfer(a(u,:));
 if(length(cur)<=1)
    live(u)=0;
    average=average+51*25;
 else
 t=f3(cur);
 live(u)=1000/(t-48);
 average=average+t;
 end
 total=total+live(u);
 end
average=average/num;
live=live/total;
[x,s]=max(live);
best=a(s,:);
if f3(transfer(best))<f3((transfer(last)))
   last=best;
end
```

```
%记录相同最优个体的持续代数
if(best==trans(bst))
    same=same+1;
else
    same=0;
end
    %利用生存概率赌轮盘法进行随机选择
    plive=cumsum(live);
for u=1:num
     ran=rand;
     tmp=find(ran<plive);
     b(u,:)=a(tmp(1),:);
end
a=b;
%随机交换 a 中个体的顺序，保证下面的配对交叉是随机的
for u=1:floor(num/2)
        ran1=randi(num);
        ran2=randi(num);
        a([ran1,ran2],:)=a([ran2,ran1],:);
end
     %配对交叉产生下一代个体
 for u=1:floor(num/4)
    b(4*u-3,22:51)=a(4*u-2,22:51);
    b(4*u-2,22:51)=a(4*u-3,22:51);
    b(4*u-1,31:51)=a(4*u,31:51);
    b(4*u,31:51)=a(4*u-1,31:51);
 end
 a=b;
  %随记变异
     for u=1:floor(rate*51*num)
     ranx=randi(num);
     rany=randi(51);
     a(ranx,rany)=1-a(ranx,rany);
     end
     a(1,:)=best;%保留最优个体,若陷入局部最优解则定向变异
     if same>=10
        for u=1:7
           ran=randi(50);
```

```
            a(1,[ran,ran+1])=a(1,[ran+1,ran]);
        end
        ran=randi(51);
        a(1,ran)=1-a(1,ran);
    end
```

%输出

```
 diary('output.txt');
 if mod(time,10)==0
   bst=transfer(best);
   fprintf('generation: %d\n',time);
   fprintf('best of the generation: %d\n');
   disp(bst);
   fprintf('value of the best:%d\n');
   disp(f3(bst));
   fprintf('average of the generation:%d\n');
   disp(average);
   end
end
   fprintf('best of the all the generations: %d\n');
   disp(transfer(last));
   fprintf('value of the best:%d\n');
   disp(f3(transfer(last)));
   diary off;
toc;
```