统计推断在数模转换系统中的应用

组号 14　　易星辉 5131619019　彭浩 5130309451

摘要：本报告以上海交通大学工程实践与科技创新 3A 的实验结果为研究数据，探究统计推断在数模模数系统转换中的应用。利用三次差值拟合方法，运用遗传算法在所有共 51 个数据点中寻找出最优的七点组合。

关键词：统计推断，三次差值拟合，遗传算法。

1. 引言

1.1 课题背景

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。

1.2 模型

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。

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）。

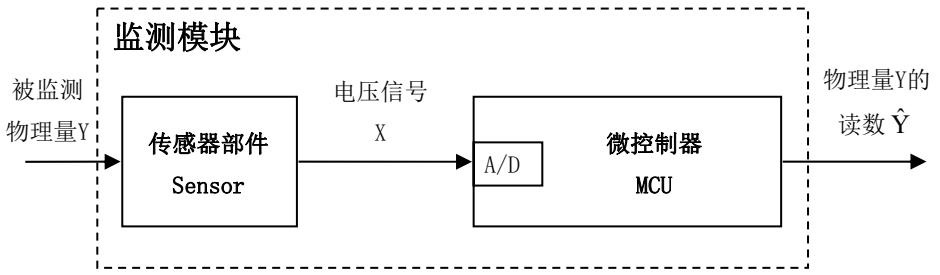


图 1 监测模块组成框图

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

$$X \in \{x_1, x_2, x_3, ..., x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, ..., 9.9, 10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i = f(x_i)$， Y 实测值记为 y_i， $i = 1, 2, 3, ..., 50, 51$。

传感部件特性

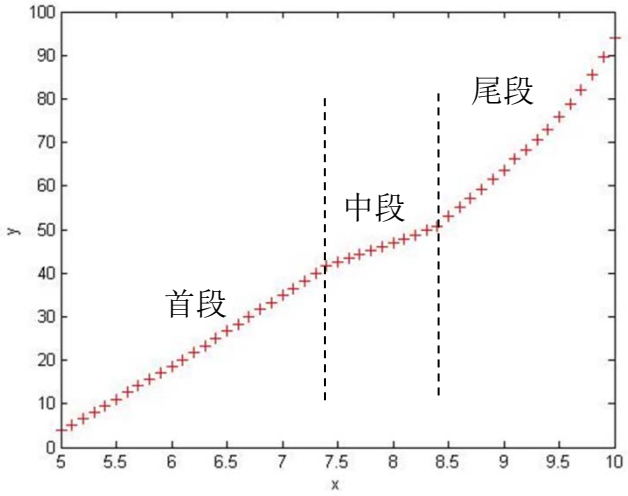


图 2 传感特性图示

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。

1.3 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。

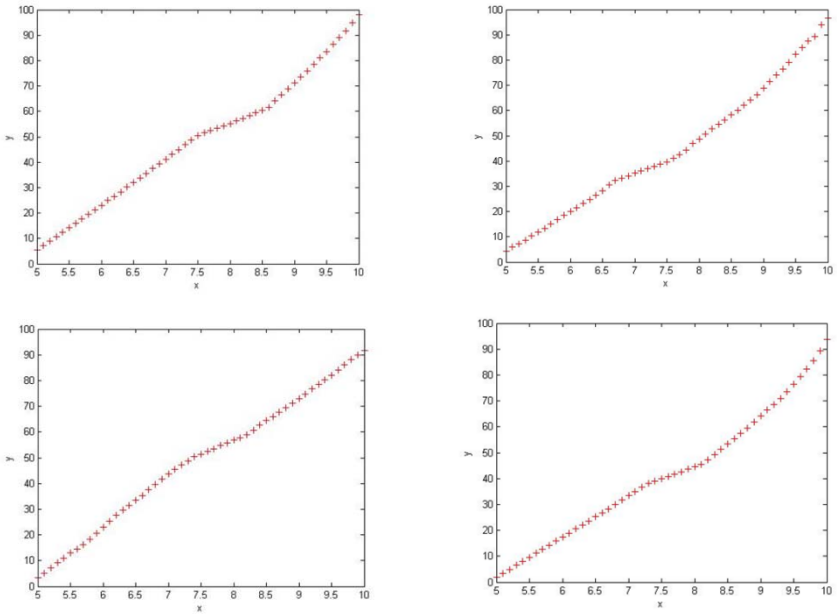


图 3 四个不同样本个体特性图示对比

1.4 测试评级函数

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

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

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。

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

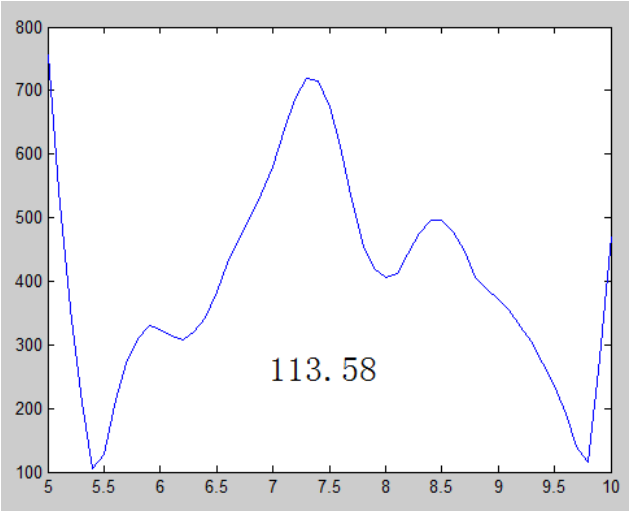
总体成本较低的校准方案，认定为较优方案。

2. 拟合方法选取

适用于本课程的拟合方法有许多，常用的有多项式拟合、指数函数拟合、插值法拟合、高斯拟合、洛伦兹拟合和傅里叶级数拟合等。

2.1 多项式拟合

多项式拟合是一个比较常见的拟合方式，由于数据的整体特性，数据大致分为单增的三段，与三次函数的大致图像很相似，所以决定采用三次多项式拟合，暂定取五点。通过 MATLAB 自带多项式拟合函数 polyfit 多次计算，得到的三次多项式拟合得到的最优成本为 113.58，这个值并不尽如人意。通过对每一点对于所有数据的误差总和进行统计，得到了下图：



由此图可以看出，成本的最大产生在数据的中断，降低成本的关键也在于中断。而产生如此大成本的原因在于，虽然数据的大致图像与三次函数很像，但是数据有明显的三段，并各具较好的线性，当三次曲线拟合时并不能很好地去迎合数据分布，从而造成较大成本，为此，我们开始尝试更换拟合方式。这里虽然与取点多少有关系，但是拟合方式是更根本的问题，我们先解决拟合方式。

2.2 样条差值拟合

在数值分析这个数学分支中，样条插值是使用一种名为样条的特殊分段多项式进行插值的形式。由于样条插值可以使用低阶多项式样条实现较小的插值误差，这样就避免了使用高阶多项式所出现的龙格现象。从多项式拟合的失败，分析认为此处可以采用三次样条差值拟合计算成本。

对于 n+1 个给定点的数据集{x_i}，我们可以用 n 段三次多项式在数据点之间构建一个三次样条。如果表示对函数 f 进行插值的样条函数，那么需要：

$$S(x)=\begin{cases}S_0(x), x\in[x_0,x_1]\\ S_1(x), x\in[x_1,x_2]\\ ....\\ S_{n-1}(x), x\in[x_{n-1},x_n]\end{cases}$$

插值特性，$S(x_i)=f(x_i)$

条相互连接，$S_{i-1}(x_i) = S_i(x_i)$， i=1,...,n-1

样

两次连续可导

$S'_{i-1}(x_i) = S'_i(x_i)$以及$S''_{i-1}(x_i) = S''_i(x_i)$， i=1,...,n-1.

由于每个三次多项式需要四个条件才能确定曲线形状，所以对于组成 S 的 n 个三次多项式来说，这就意味着需要 4n 个条件才能确定这些多项式。但是，插值特性只给出了 n + 1 个条件，内部数据点给出 n + 1 - 2 = n - 1 个条件，总计是 4n - 2 个条件。我们还需要另外两个条件，根据不同的因素我们可以使用不同的条件。

通俗一点讲，如果是去六点进行三次样条差值，则取相邻的四点得到的三次曲线就作为中间两点见的拟合曲线。由于成本与取点个数相关，这里并不能分析出取点多少对于成本结果的影响，故在后面实际计算时进行比较。

3. 遗传算法

3.1 问题起源

本课程设置本质上是一个搜索问题，如果对全部数据进行搜索，所需搜索的范围巨大，为种组合，穷举搜索理论上将花费大量时间，实际上是不可能实现的 ，属于所谓的 NP 问题。所以我们决定通过遗传算法优化取点方案，通过拟合得到较优的结果。

3.2 遗传算法

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。它是由美国的 J.Holland 教授 1975 年首先提出，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。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。

对于一个求函数最大值的优化问题(求函数最小值也类同)，一般可以描述为下列数学规划模型：式中 x 为决策变量，式 2-1 为目标函数式，式 2-2、2-3 为约束条件，U 是基本空间，R 是 U 的子集。满足约束条件的解 X 称为可行解，集合 R 表示所有满足约束条件的解所组成的集合，称为可行解集合。

$$\begin{cases} \max f(X) & 2-1 \\ x \in R & 2-2 \\ R \subset U & 2-3 \end{cases}$$

遗传算法也是计算机科学人工智能领域中用于解决最优化的一种搜索启发式算法，是进化算法的一种。这种启发式通常用来生成有用的解决方案来优化和搜索问题。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优，而不能达到全局最优。

3.3 一般步骤

遗传算法的基本运算过程如下：

a)初始化：设置进化代数计数器 t=0，设置最大进化代数 T，随机生成 M 个个体作为初始群体 P(0)。

b)个体评价：计算群体 P(t)中各个个体的适应度。

c)选择运算:将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

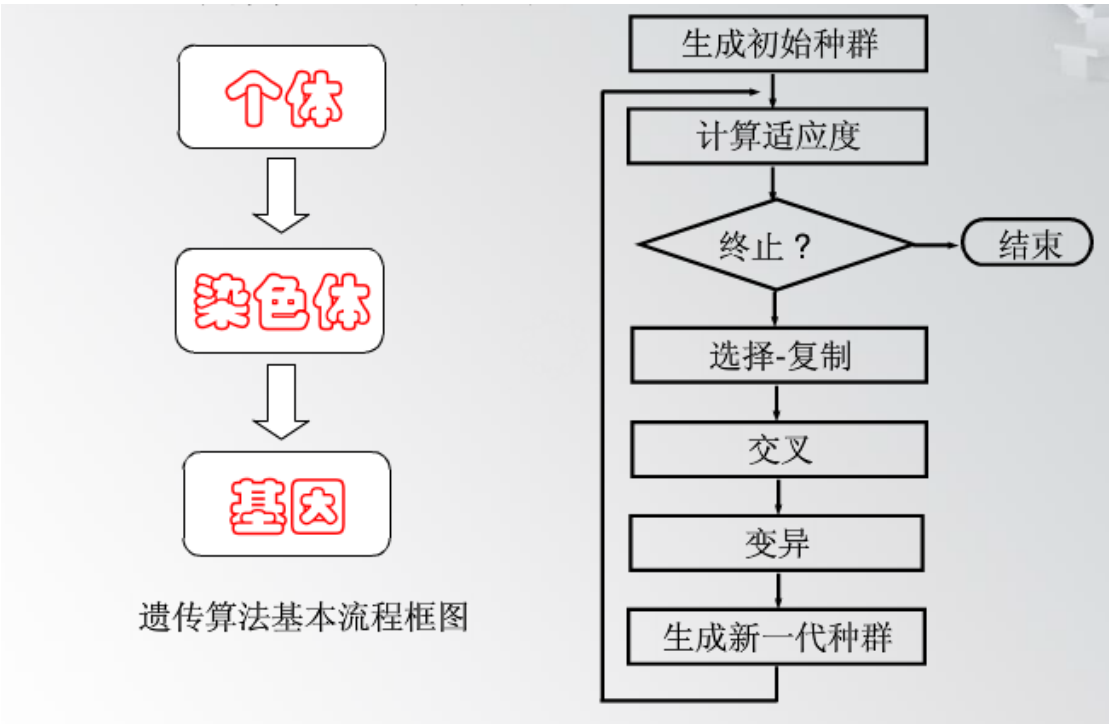
d)交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

e)变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。

群体 P(t)经过选择、交叉、变异运算之后得到下一代群体 P(t+1)。

f)终止条件判断:若 t=T,则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

流程图如下



3.4 遗传算法的具体实现

虽然 MATLAB 有其自带的遗传算法库，但它不便于我们人为地，根据数据特点增加一些逻辑判断，所以决定自编，大致步骤如下（详见代码）：

1.初始化：

随机生成 100 个个体作为初始种群，每个种群有取样点多个。

2.遗传算法主体：

（1）计算适应度：利用个体进行三次差值拟合，通过给定的成本计算方式成本，将最大成本个体作为基准 0，得到其他成本与其差值（正值），然后每一点的适应度为这一点与其之前所有点的差值之和与总差值的比值，这样做是为了淘汰最差个体。

（2）选择复制：通过随机生成的一个 0~1 之间的小数，求其对应的区段所对应的个体，选择其生存下来。

（3）交叉变异：交叉变异进行 0~101 之间的一个随机数的次数。交叉变异进行时，是将选中的个体数值转化为二进制，然后像 DNA 一样的一段一段的交换，一个值一个值的取反得到新的种群。

（4）非法数据的处理：交叉变异后会产生超过 51 数值，或者重复取点的非法情况。对于超过 51 数值时，我们将其重新更改为 102-x，因为通过对取点结果的成本统计，后段会产生较大的成本，所以在点的后段取更多的点有利于降低成本。对于重复的点，我们将会让其个体与当前记录的最优方案再次进行交叉变异，有更大的希望产出好的取点方案。

2.输出处理：

每一代的最优取点都会被输出，在最后会输出整个过程中的最优取点。

4. 算法结果及结果分析

拟合方式	取点数量	最优取点方案	成本	运行代数
三次多项式拟合	5	5,15,28,39,49	113.58	120
三次样条差值拟合	6	4,12,21,31,43,47	98.35	120
三次样条差值拟合	7	2,10,20,29,34,44,50	94.00	120
三次样条差值拟合	8	1,9,15,24,29，34,45,50	104.26	120

通过上面的计算我们可以看出：取点分布比较均匀的解优于取点分布不均匀的解；三次样条差值拟合的成本均低于样条差值，而样条差值时，取点数虽然能减小没点的误差，但是也会增大总成本，在此间去一个平衡，从结果看以取点方案（2，10,20,29,34,44,50）为最佳。

5. 致谢

刚接触课题时我们非常茫然，以前没接触 matlab 编程，感到无从下手。但是通过对遗传算法和模拟退火算法的了解和学习，我们逐渐对课题内容熟悉起来，即使其中遇到了诸多不解的地方，我们也得到了学长和同学们的帮助而突破难关，感谢他们能伸出援手，也很感谢袁老师在面谈时给我们组的指导和建议，这让我们在算法方面优化了许多，虽然我们的算法还存在一些不足，但是这种不断推敲、认真钻研的精神是我们在统计推断课上学到的最宝贵的财富。

6. 参考文献

1. 上海交通大学统计推断提供的课程 ppt
2. http://blog.csdn.net/b2b160/article/details/4680853/非常好的遗传算法的例子
3. http://www.360doc.com/content/11/0130/15/991597_89950992.shtml 遗传算法学习心得
4. http://my.oschina.net/u/1412321/blog/192454 遗传算法
5. http://my.oschina.net/u/1412321/blog/192454 遗传算法
6. http://zh.wikipedia.org/wiki/%E6%A0%B7%E6%9D%A1%E6%8F%92%E5%80%BC 维基百科 样条差值
7. http://zh.wikipedia.org/zh-cn/%E6%9B%B2%E7%B7%9A%E6%93%AC%E5%90%88 曲线拟合

7. 附录：matlab 程序

主程序：main.m
子程序：cross_variation.m（交叉变异），getcost.m（获得成本），getfitness.m（计算适应度），selection.m（选择），refresh.m（再次交叉变异）

main.m

```
global n yyy ee minp min t fit pos ros
format long g
%------------------------------读入并初始化----------------------
yyy=zeros(1,51*469*2);
inin=dlmread('20141010dataform.csv');
[l,r]=size(inin);
for i=1:l
    for j= 1:r
        yyy((i-1)*51+r)=inin(i,j);
    end;
end;
pos=7;          %pos 取点数
ros=100;        %ros 种群数
ee=zeros(ros+1,pos+1);
%前七位为取点编号，第八位为 cost 前 ros 行为现有种群，最后一行为历史最小 cost 种群
i=1;
ee(ros+1,pos+1)=327670;
while i<=ros
    j=1;
    while j<=pos
        q=1;
        while q==1
```

```
            ee(i,j)=randi(51);
            qq=1;
            for p=1:j-1
                if ee(i,j)==ee(i,p)
                    qq=0;
                end;
            end;
            if qq==1 q=0; end;
        end;
        j=j+1;
    end;
    i=i+1;
end;
minp=0;
min=327670;
getcost;
i=1;
while i<=ros
    if ee(i,pos+1)<ee(ros+1,pos+1)
        ee(ros+1,:)=ee(i,:);
    end;
    if ee(i,pos+1)<min
        min=ee(i,pos+1);
        minp=i;
    end;
    i=i+1;
end;
fprintf('   代数                              最优种群
mincost\n');
fprintf('%5.f          %5.f,%5.f,%5.f,%5.f,%5.f,%5.f,%5.f           %
5.1f\n',0,ee(minp,1),ee(minp,2),ee(minp,3),ee(minp,4),ee(minp,5),ee(m
inp,6),ee(minp,7),min);
t=[0,0,0,0];
fit=327670*ones(1,ros);
%-------------------------遗传算法搜索取点方法-------------------
n=51;
num=1;
%fprintf('  代数           最优种群             mincost\n');
while num<=120
    %计算适应度
    getfitness;
    %选择
    selection;
    %交叉变异
```

```
    cross_variation;
    %更新算 cost
    minp=0;
    min=327670;
    getcost;
    i=1;
    while i<=ros
        if ee(i,pos+1)<ee(ros+1,pos+1)
            ee(ros+1,:)=ee(i,:);
        end;
        if ee(i,pos+1)<min
            min=ee(i,pos+1);
            minp=i;
        end;
        i=i+1;
    end;
    %输出
    fprintf('   代数                              最优种群
mincost\n');

fprintf('%5.f          %5.f,%5.f,%5.f,%5.f,%5.f,%5.f,%5.f           %
5.1f\n',num,ee(minp,1),ee(minp,2),ee(minp,3),ee(minp,4),ee(minp,5),ee
(minp,6),ee(minp,7),min);
    num=num+1;
end;
minp=ros+1;
min=ee(minp,pos+1);
fprintf('
best        %5.f,%5.f,%5.f,%5.f,%5.f,%5.f,%5.f           %5.1f\n',ee(
minp,1),ee(minp,2),ee(minp,3),ee(minp,4),ee(minp,5),ee(minp,6),ee(min
p,7),min);
```

cross_variation.m（交叉变异）

```
global ee pos ros n m
%----------------cross----------------
s=randi(ros);
while s>0
    x1=randi(ros); x2=randi(ros);
    while x1==x2
        x2=randi(ros);
    end;
    y1=randi(pos); y2=randi(pos);
    poss=randi(pos);
    ex1=[0,0,0,0,0,0]; ex2=[0,0,0,0,0,0];
```

```
    a=ee(x1,y1); w=1;
    while a>0
        ex1(w)=mod(a,2);
        a=(a-mod(a,2))/2;
        w=w+1;
    end;
    a=ee(x2,y2); w=1;
    while a>0
        ex2(w)=mod(a,2);
        a=(a-mod(a,2))/2;
        w=w+1;
    end;
    for i=poss:6
        tmp=ex1(i); ex1(i)=ex2(i); ex2(i)=tmp;
    end;
    ee(x1,y1)=ex1(1)+ex1(2)*2+ex1(3)*4+ex1(4)*8+ex1(5)*16+ex1(6)*32;
    if ee(x1,y1)>51
        ee(x1,y1)=102-ee(x1,y1);
    end;
    ee(x2,y2)=ex2(1)+ex2(2)*2+ex2(3)*4+ex2(4)*8+ex2(5)*16+ex2(6)*32;
    if ee(x2,y2)>51
        ee(x2,y2)=102-ee(x2,y2);
    end;
    s=s-1;
end;
%-----------------------------variation-------------------------
s=randi(ros);
while s>0
    x1=randi(ros);
    x2=randi(pos);
    ex1=[0,0,0,0,0,0];
    a=ee(x1,x2); w=1;
    while a>0
        ex1(w)=mod(a,2);
        a=(a-mod(a,2))/2;
        w=w+1;
    end;
    exc=randi(6);
    if ex1(exc)==0
        ex1(exc)=1;
    else ex1(exc)=0;
    end;
    ee(x1,x2)=ex1(1)+ex1(2)*2+ex1(3)*4+ex1(4)*8+ex1(5)*16+ex1(6)*32;
    if ee(x1,x2)>51
```

```
        ee(x1,x2)=102-ee(x1,x2);
    end;
    s=s-1;
end;
%-------------------------若处理重复点数及 0 点--------------------
for i=1:ros
    while ee(i,1)==0
        n=i; m=1;
        refresh;
    end;
    for j=2:pos
        k=1;
        while k<j
            if (ee(i,j)==ee(i,k) | ee(i,j)==0)
                q=1;
                while q==1
                    n=i; m=j;
                    refresh;
                    qq=1;
                    for p=1:j-1
                        if ee(i,j)==ee(i,k)
                            qq=0;
                        end;
                    end;
                    if qq==1 k=0; q=0; end;
                end;
            end;
            k=k+1;
        end;
    end;
end;
```

getcost.m（获得成本）

```
global yyy ee t pos ros
yy=zeros(469,51);
for i=1:469
    for j=1:51
        yy(i,j)=yyy((i-1)*102+51+j);
    end;
end;
for k=1:ros
    cost=0;
    for g=1:469
        x=0*ones(1,pos); y=0*ones(1,pos);
```

```
        for i=1:pos
            x(i)=ee(k,i)/10+4.9;
            y(i)=yy(g,ee(k,i));
        end;
        t=zeros(1,51);
        xj=[5.0:0.1:10];
        t=spline(x,y,xj);
        errabs=abs(yy(g,:)-t);
        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);
        for p=1:51
            cost=cost+sij(p);
        end;
    end;
    ee(k,pos+1)=cost/469+pos*12;
end
```

getfitness.m（计算适应度）

```
global ee fit pos ros
max=ee(1,pos+1);
for i=2:ros
    if max<ee(i,pos+1) max=ee(i,pos+1);
    end;
end;
for i=1:ros
    fit(i)=max-ee(i,pos+1);
end;
```

selection.m（选择）

```
global ee fit eee pos ros
xx=rand(ros);
val=0;
for i=1:ros
    val=val+fit(i);
    if i>1 fit(i)=fit(i)+fit(i-1); end;
    tt(i)=xx(i)*val;
```

```
end;
eee=zeros(ros+1,pos+1);
eee(ros+1,:)=ee(ros+1,:);
for i=1:ros
    q=1;
    while (q<200)
        if tt(i)<fit(q)
            eee(i,:)=ee(q,:);
            q=201;
        end;
        q=q+1;
    end;
end;
ee=eee;
```

refresh.m（再次交叉变异）

```
global ee pos ros n m
x1=ros+1; x2=n;
    y1=randi(pos); y2=m;
    poss=randi(6);
    ex1=[0,0,0,0,0,0]; ex2=[0,0,0,0,0,0];
    a=ee(x1,y1); w=1;
    while a>0
        ex1(w)=mod(a,2);
        a=(a-mod(a,2))/2;
        w=w+1;
    end;
    a=ee(x2,y2); w=1;
    while a>0
        ex2(w)=mod(a,2);
        a=(a-mod(a,2))/2;
        w=w+1;
    end;
    for ss=poss:6
        ex2(ss)=ex1(ss);
    end;
    ee(x2,y2)=ex2(1)+ex2(2)*2+ex2(3)*4+ex2(4)*8+ex2(5)*16+ex2(6)*32;
    if ee(x2,y2)>51
        ee(x2,y2)=102-ee(x2,y2);
    end;
```