

统计推断在 ADC 系统中的应用

组号：46 组员：宋凯敏 F1203021 5120309575 葛波利 F1203021 5120309578

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 ADC 系统中的应用》的课程论文。本报告以 ftp 的实验结果为研究数据，探讨统计推断在数模模数转换中的应用。我们运用 Matlab 数学软件并使用统计推断的方法，通过 469 组数据对系统的输入输出关系进行研究。

关键词：统计推断、Matlab、特征点、曲线拟合、模拟退火算法、遗传算法。

Application of Statistical Inference in ADC Inverting System

ABSTRACT: This article is for Course Design-Application of Statistical Inference in ADC System, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. This report takes the results in ftp as research data, to explore the application of statistical inference in the digital to analog conversion. We use statistical inference, in combination with Matlab, to research into the relationship between the input and output, based on the given data.

Key words: statistical inference、Matlab、Feature points、curve fitting、simulated annealing algorithm, genetic algorithm.

1 引言

本课题研究的内容为输入与输出之间的函数关系，由于两者并不存在确定的函数关系，因而不仅需要通过研究大量的数据（469 组实验数据）还要寻找适合的拟合函数、筛选相关数据，运算量较大，并且较为难以实现。因此，本课题旨在找出我们能够找出的误差较小的拟合函数，是的所有样本的共同点在这一函数中得以体现出来。

2 课程问题

基本问题的最终目标是找到 6 个特征点，是根据这 6 个点拟合的数据按照提供的函数计算后，所有的样本的越低越好。由题条件，成本的结果按照以下规则：

(S_i 为第 i 组的得分， $Y_{i,j}$ 表示第 i 组数据的第 j 点测得的输出， $\hat{Y}_{i,j}$ 表示根据拟合函数

求得的输出)

$$s_{i,j} = \begin{cases} 0 & \text{if } |\hat{y}_{i,j} - y_{i,j}| \leq 1 \\ 1 & \text{if } 1 < |\hat{y}_{i,j} - y_{i,j}| \leq 2 \\ 2 & \text{if } 2 < |\hat{y}_{i,j} - y_{i,j}| \leq 3 \\ 4 & \text{if } 3 < |\hat{y}_{i,j} - y_{i,j}| \leq 5 \\ 10 & \text{if } |\hat{y}_{i,j} - y_{i,j}| > 5 \end{cases}$$

$$S_i = \sum_{j=1}^{51} s_{i,j} + 20N_i$$

最终的成本为

$$C = \frac{1}{M} \sum_{i=1}^M S_i$$

最终的总成本为

而第一个点与第六个点的选取，我们固定为第 1 点与第 51 点。

3 关于函数拟合

输入和输出之间虽然没有很明显的线性关系，但是其依旧具有较为平滑的曲线关系。

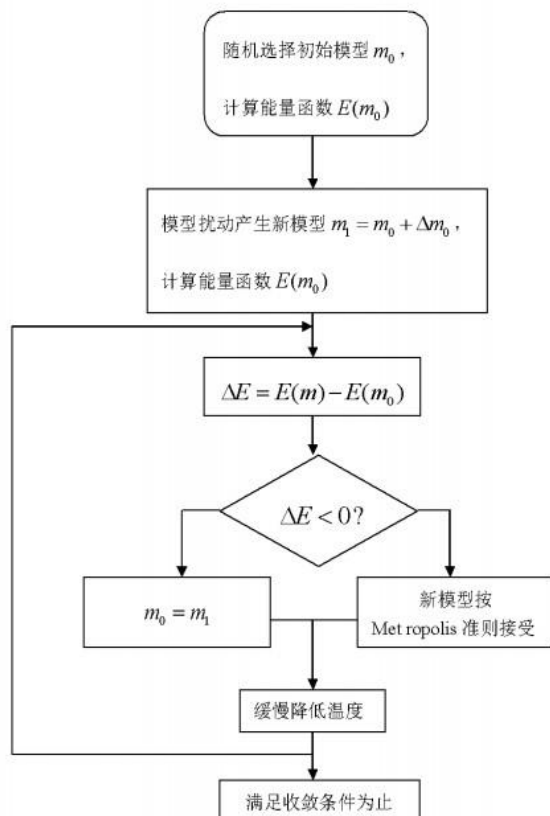
因此，我们首先尝试采用启发式 heuristic 搜索算法中的模拟退火算法对 6 个事前观测点进行选取以及分段多项式拟合来对各段函数进行拟合。

3.1 模拟退火算法

3.1.1 模拟退火算法简介

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。

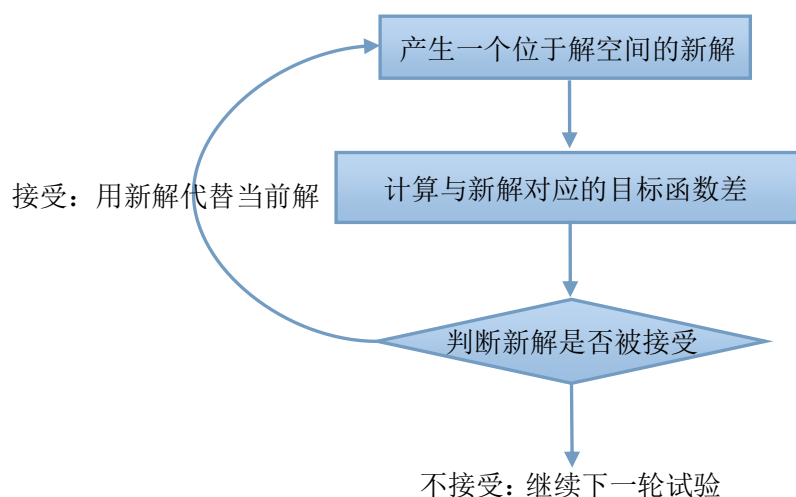
模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表(Cooling Schedule)控制，包括控制参数的初值 t 及其衰减因子 Δt 、每个 t 值时的迭代次数 L 和停止条件 S 。



3.1.1 模拟退火算法在本报告中的应用

模拟退火算法可以分解为解空间、目标函数和初始解三部分。

我们根据模拟退火算法，在本次报告中新解的产生和接受可以用一下流程图表示：



关于以上流程图的解释：

- 1、生成函数的初解：初解的生成借用随机函数在 469 组的数据中产生。
- 2、判断新解是否被接受：应用 Metropolis 准则：若 $\Delta E < 0$ 则接受为新的当前解，否则以概率 $\exp(-\Delta E/T)$ 接受作为新的当前解。
- 3、新解生成函数：扰动的产生依旧借助于随机函数在原数据组中进行随机取样，每次改变

4 个点中的一个。

（对于这观测点的取值范围我定为： ）

序号	取值范围
1	1
2	7~27
3	13~33
4	19~39
5	25~45
6	51

4、终止条件：目标函数的成本函数结果小于等于 130（其中 120 为测定成本，10 为误差成本）

3.2 分段多项式拟合

由于整体系统特征与其电路系统中晶体管元件特性密切相关，晶体管工作特性可分为前后衔接的连续区间：截止区、线性区和饱和区，在跨越区间是物理特性是连续而非跳跃间断的；我们假定整体系统特性也会呈现前后衔接的 N 个区间，每个徐建适合分别用一段多项式曲线进行拟合，前后区间对应的多项式曲线在区间衔接处取值连续。

3.2.1 区间分段点的选取

我通过各个点处所对应的平均曲率变化来寻找区间分段点。

我通过对数据库中若干组解的取平均值后，可以得到近似的曲率的区间分为一个分段，这样我们就可以在间隔为 6 的数组中寻到到有限个分段区间，进而可以得到相对精确的分段拟合函数，如之前退火算法中介绍过的：

序号	取值范围
1	1
2	7~27
3	13~33
4	19~39
5	25~45
6	51

3.2.2 每个区间分段中函数的拟合

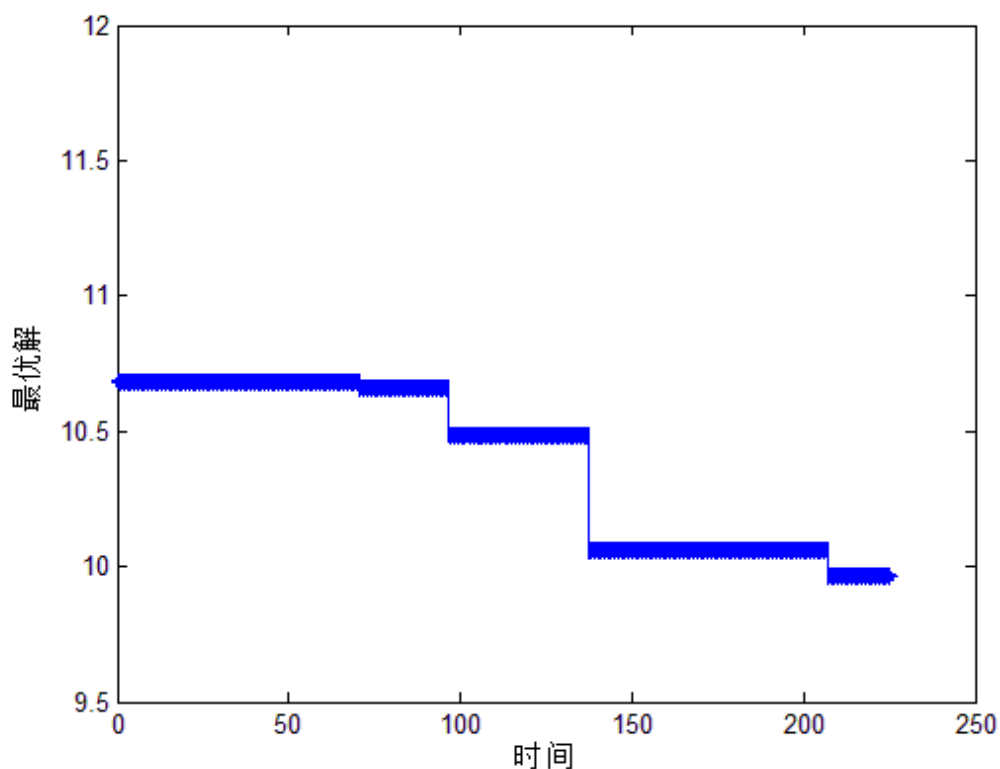
一般意义而言，最佳的拟合函数是个不可能达到的要求，部分样本自身数据的误差会导致拟合效果极差，基本不可能出现对于所有的数据点都极为符合的情况。往届同学的研究也反映了这一点，在此条件下，连续多次的拟合尝试都没有能达到最佳的效果。在后期的继续尝试中，我们尝试了基于最小二乘法的多项式拟合，在每一个分段当中，有的分段线性较好的可以撇除离散点对其拟合的影响，而对于曲线类型的分段，我们原本预期将进行三次以及四次多项式拟合的尝试。

最终发现，三次的拟合就能达到目标的效果，相比于二次的拟合有了极为明显的提升，相对算法也较为简单。因此我们对于中间的分段进行三次的拟合，应用与中间的分段，而对于两边的分段仍旧应用二次拟合。（如下所示）

分段	拟合方式
1——2	通过 1,2,3 点的二次拟合
2——3	通过 1,2,3,4 点的三次拟合
3——4	通过 2,3,4,5 点的三次拟合
4——5	通过 3,4,5,6 点的三次拟合
5——6	通过 4,5,6 点的二次拟合

最终的出最优点为：1、9、21、27、32、44、51（成本平均值为 9.8290）

最优解分布如图所示：



3.3 遗传算法

3.3.1 遗传算法简介

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是从代表问题可能潜在的解集的一个种群开始的，而一个种群则由经过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体。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，逐代演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度大小选择个体，并借助于自然遗传学的遗传算子进行组合交叉和变异，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生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，就可以作为问题近似最优解。^[4]

3.3.2 遗传算法在本课程中的应用

遗传算法的基本运算过程如下：

1)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 $P(0)$ 。

2)个体评价：计算群体 $P(t)$ 中各个个体的适应度。

3)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。

4)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。

5)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。

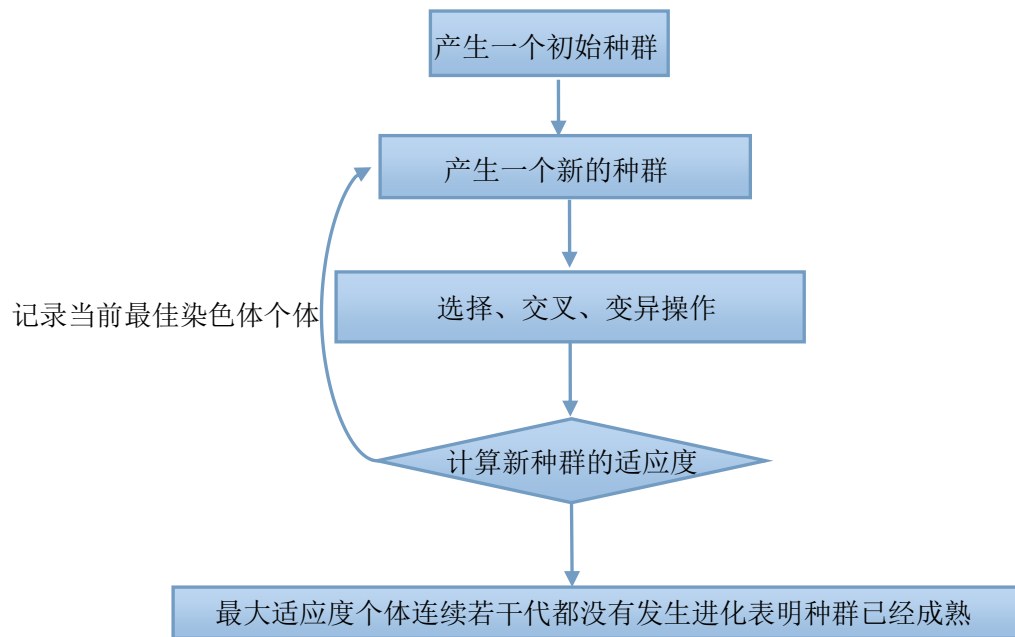
群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ 。

6)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。

我们根据遗传算法，在本次报告中新解的产生和接受可以用一下流程图表示：

确定取值范围、运算精度、初始种群大小、最大代数、交配变异概率





经过讨论，我们将一维自变量的取值范围定为 $[2, 52]$ ，运算精度定为 0.0001，初始种群大小为 50，最大代数为 100，交配概率为 0.50，变异概率为 0.09。最大适应度个体连续 6 代都没有发生进化表明种群已经成熟。

最终得出的最优个体为：1、10、21、28、33、44、51

4 参考文献

- [1] 上海交大电子工程系，统计推断在 ADC 系统中的应用课程讲义
- [2] Matlab 在数学建模中的应用 卓金武 北京航空航天大学出版社
- [3] 模拟退火算法机理研究 陈华根，吴建生，王家林，等。同济大学学报：自然科学版，2004,32（6）：802-805
- [4] 百度百科

5 附录（代码）

5.1 退火算法

```
function main()  
clc;  
clear all;  
close all;  
global data  
global data_size  
tic;  
data = dlmread('20141010dataform.csv')  
data_size =size( data );  
select2(data);
```

```
%-----  
function select2(data)  
si=[ 1    11    21    31    41    51 ];  
snew=si;  
smax=zeros(6);  
[n,m]=size(data);  
D=zeros(n/2,m);  
U=D;  
Ui=U;  
score=zeros(n/2,1);  
k=1:2:n;  
D=data(k,:);  
k=2:2:n;  
U=data(k,:);  
T=5000;  
Tmin=0.05;  
r=0.95;  
fenold=targetfun(si);  
disp(fenold);  
scoremin=fenold;  
times=1;  
  
%开始模拟退火%
```



```

tic

while (T>Tmin)
    change=round((rand(1,6)-0.5)*2);
    %   if sum(abs(change))>4
    %       continue
    %   end
    snew=si+change;
    if
        (sum(snew==0)||sum(snew==52)||abs(snew(1)-snew(2))<6||abs(snew(2)-snew(3))<6||...
        abs(snew(2)-snew(3))<6||abs(snew(3)-snew(4))<6||abs(snew(4)-snew(5))<6||...
        abs(snew(5)-snew(6))<6)
    continue
    end
    disp(T);
    disp(snew);
    disp(scoremin);
    fennew=targetfun(snew);
    ds=fennew-fenold;
    if ds>0
        si=snew;
        fenold=fennew;
        if fennew<scoremin
            smax=si;
            scoremin=fennew;
        end
    else
        if exp(1000*ds/T)>rand
            si=snew;
            fenold=fennew;
        end
    end
    T=T*r;
    a(times)=scoremin;
    times=times+1;

end
times=times-1;
toc
disp(smax)
disp(fenold)

```

```

disp('模拟退火结束')

figure(1);
hand1=plot(1:times,a);
set(hand1,'linestyle','-','linewidth',1.8,'marker','*','markersize',6
)
xlabel('时间');ylabel('最优解');
hold on;

function seln=selection(population,cumsump);
%从种群中选择两个个体
for i=1:2
    r=rand; %产生一个随机数
    prand=cumsump-r;
    j=1;
    while prand(j)<0
        j=j+1;
    end
    seln(i)=j; %选中个体的序号
end

%-----
global Data
global Data_size
global flag
Data = dlmread('20141010dataform.csv')
Data_size =size( Data );

individual=[1 6 20 36 45 51]
individual=sort(individual);
%解决相同元素问题
pc=1;
while pc>0
    pc=-1;
    individual=sort(individual);
    for i=2:6
        flag=round(individual(i)-individual(i-1));
        if flag==0
            individual(i)=round(individual(i)+1);
            if round(individual(i))==52
                individual(i)=1;
                pc=1;
                individual=sort(individual);
            end
        end
    end
end

```

```

        break
    end
    else
        individual(i)=individual(i);
    end
end
end

individual=sort(individual);

Score = zeros( 1, Data_size(1)/2 );
for i = 1 : Data_size(1)/2
    Y = interp1( Data( 2*i-1 , individual(1, : ) ),Data( 2*i,
individual(1, : ) ),Data( 2*i-1 ,2:51 ),'spline');
    dert_Y = abs( Y(:, :) - Data( 2*i, 2:51 ) );
    for j = 1:( Data_size(2) - 2 )
        if dert_Y( j ) <= 1
            Score( 1, i ) = Score( 1, i ) ;
        elseif dert_Y( j ) <= 2
            Score( 1, i ) = Score( 1, i ) + 1;
        elseif dert_Y( j ) <= 3
            Score( 1, i ) = Score( 1, i ) + 2;
        elseif dert_Y( j ) <= 5
            Score( 1, i ) = Score( 1, i ) + 4;
        else
            Score( 1, i ) = Score( 1, i ) + 10;
        end
    end
end
end
y=mean(Score(1,:));

%-----
function [Fitvalue,cumsump]=fitnessfun(population);
global BitLength
global boundsbegin
global boundsend

popsize=size(population,1); %有 popsize 个个体

for b=1:popsize
    x=transform2to10(population(b,:));
    xx=round(boundsbegin+x*(boundsend-boundsbegin)/(power(2 ,7)-1));%
    二进制转十进制，然后再转化为[1,51]区间的实数

```

```

Fitvalue(b)=targetfun(xx);
end

```

%给适应度函数加上一个大小合理的数以便保证种群适应值为正数

```
Fitvalue=Fitvalue'+230;
```

%计算选择概率

```
fsum=sum(Fitvalue);
```

```
Pperpopulation=Fitvalue/fsum;
```

%计算累积概率

```
cumsump(1)=Pperpopulation(1);
```

```
for i=2:popsize
```

```
    cumsump(i)=cumsump(i-1)+Pperpopulation(i);
```

```
end
```

```
cumsump=cumsump';
```

```
%
```

```
function c=fitting_try(x,y,m)
```

%x 为数据点的横坐标

%y 为数据点的纵坐标

%m 为多项式最高次数

%返回结果为 F，拟合多项式函数

```
n=length(x);
```

```
b=zeros(1,m+1);
```

```
f=zeros(n,m+1);%f 为正规方程的系数,初始为 0_try
```

```
for k=1:m+1
```

```
    f(:,k)=x'.^(k-1);
```

```
end
```

```
a=f'*f;
```

```
b=f'*y';
```

%解方程,得到多项式由高到低的系数所构成的向量 c

```
c=a\b;
```

```
c=flipud(c);%翻转
```

%以上用了最小二乘法

```
end
```

```
%
```

```
function [population]=makesureDifferent (population);
```

```
popsize=size (population,1);
```

```

for i=1:popsize
    individual=population(i,:);
    for i=1:6
        flag=individual(i)-individual(
%-----
function y=targetfun(s); %目标函数
global data
global data_size
global flag
s=sort(s);
%解决相同元素问题
for i=2:6
    for j=2:i
        flag=s(i)-s(j-1);
        if flag==0
            s(i)=s(i)+(i-j+1);
        end
    end
end
end

s=sort(s);

Score = zeros( 1, data_size(1)/2 );
for i = 1 : data_size(1)/2
    Y = interp1( data( 2*i-1 , s(1, : ) ),data( 2*i,
s(1, : ) ),data( 2*i-1 ,2:51 ),'spline');
    dert_Y = abs( Y(:, :) - data( 2*i, 2:51 ) );

for j = 1:( data_size(2) - 2 )
    if dert_Y( j ) <= 1
        Score( 1, i ) = Score( 1, i ) ;
    elseif dert_Y( j ) <= 2
        Score( 1, i ) = Score( 1, i ) + 1;
    elseif dert_Y( j ) <= 3
        Score( 1, i ) = Score( 1, i ) + 2;
    elseif dert_Y( j ) <= 5
        Score( 1, i ) = Score( 1, i ) + 4;
    else
        Score( 1, i ) = Score( 1, i ) + 10;
    end
end
end
end

```

```
y=mean(Score(1,:));
```

5.2 遗传算法

```
clc;
clear all;
close all;
tic;
global BitLength
global boundsbegin
global boundsend
global Data
global Data_size
Standard = 94.0;
Data = xlsread('数据表格.xls','C2:BC1089');
Data_size =size( Data );
bounds=[2 52];% 一维自变量的取值范围
%precision=0.0001; %运算精度
boundsbegin=bounds(:,1);
boundsend=bounds(:,2);
%计算如果满足求解精度至少需要多长的染色体
BitLength=56;
popsize=50; %初始种群大小
Generationnmax=100; %最大代数
pcrossover=0.50; %交配概率
pmutation=0.09; %变异概率
%产生初始种群
population=round(rand(popsize,BitLength));
%计算适应度,返回适应度 Fitvalue 和累积概率 cumsum
[Fitvalue,cumsum]=fitnessfun(population);
Generation=1;
while Generation<Generationnmax+1
    for j=1:2:popsize
        %选择操作
        seln=selection(population,cumsum);
        %交叉操作
        scro=crossover(population,seln,pcrossover);
        scnew(j,:)=scro(1,:);
        scnew(j+1,:)=scro(2,:);
        %变异操作
        smnew(j,:)=mutation(scnew(j,:),pmutation);
```

```

        smnew(j+1,:)=mutation(scnew(j+1,:),pmutation);
    end
    population=smnew; %产生了新的种群
    %计算新种群的适应度
    [Fitvalue,cumsump]=fitnessfun(population);
    %记录当前代最好的适应度和平均适应度
    [fmax,nmax]=max(Fitvalue);
    fmean=mean(Fitvalue);
    ymax(Generation)=fmax-230;
    ymean(Generation)=fmean-230;
    %记录当前代的最佳染色体个体
    x=transform2to10(population(nmax,:));
    %自变量取值范围是[-2 2],需要把经过遗传运算的最佳染色体整合到[-2 2]区间
    xx=round(boundsbegin+x*(boundsend-boundsbegin)/(power(2,8)-1));
    xmax(Generation,:)=xx;
    sp1=num2str(xx);
    sp2=num2str(fmax-230);
    disp(strcat('Generation','=',sp1));
    disp(strcat('the maximum value','=',sp2));
    tmp=Generation;
    if abs(ymax(Generation))>Standard

        tmp=Generation;
        Generation=Generationnmax+1;
        break
    end
    Generation=Generation+1

end
Generation=tmp+1;
sp1=num2str(xx);
sp2=num2str(fmax-230);
disp(strcat('the maximum generation','=',sp1));
disp(strcat('the maximum value','=',sp2));
Generation=Generation-1;
Bestpopulation=xx
Besttargetfunvalue=targetfun(xx)
%绘制经过遗传运算后的适应度曲线。一般地,如果进化过程中种群的平均适应度与最大适
%应度在曲线上有相互趋同的形态,表示算法收敛进行得很顺利,没有出现震荡;在这种前
%提下,最大适应度个体连续若干代都没有发生进化表明种群已经成熟。
figure(1);
hand1=plot(1:Generation,ymax);
set(hand1,'linestyle','-','linewidth',1.8,'marker','*','markersize',6)
hold on;

```

```

hand2=plot(1:Generation,ymean);
set(hand2,'color','r','linestyle','-','linewidth',1.8,...
'marker','h','markersize',6)
xlabel('进化代数');ylabel('最大/平均适应度');xlim([1 Generationnmax]);
legend('最大适应度','平均适应度');
box off;hold off;
toc;
%子程序：新种群选择操作，函数名称存储为 selection.m
function seln=selection(population,cumsump);
%从种群中选择两个个体
for i=1:2
    r=rand; %产生一个随机数
    prand=cumsump-r;
    j=1;
    while prand(j)<0
        j=j+1;
    end
    seln(i)=j; %选中个体的序号
end
%子程序：新种群交叉操作,函数名称存储为 crossover.m
function scro=crossover(population,seln,pc);
BitLength=size(population,2);
pcc=IfCroIfMut(pc); %根据交叉概率决定是否进行交叉操作，1 则是，0 则否
if pcc==1
    chb=round(rand*(BitLength-2))+1; %在[1,BitLength-1]范围内随机产生一个交叉位
    scro(1,:)=[population(seln(1),1:chb) population(seln(2),chb+1:BitLength)];
    scro(2,:)=[population(seln(2),1:chb) population(seln(1),chb+1:BitLength)];
else
    scro(1,:)=population(seln(1),:);
    scro(2,:)=population(seln(2),:);
end
%子程序：计算适应度函数，函数名称存储为 fitnessfun
function [Fitvalue,cumsump]=fitnessfun(population);
global BitLength
global boundsbegin
global boundsend

popsize=size(population,1); %有 popsize 个个体

for b=1:popsize
    x=transform2to10(population(b,:));
    xx=round(boundsbegin+x*(boundsend-boundsbegin)/(power(2 ,8)-1));%二进制转十进制，
    然后再转化为[1,53]区间的实数

```



```

        Fitvalue(b)=targetfun(xx);
end

```

%给适应度函数加上一个大小合理的数以便保证种群适应值为正数

```
Fitvalue=Fitvalue'+230;
```

%计算选择概率

```
fsum=sum(Fitvalue);
```

```
Pperpopulation=Fitvalue/fsum;
```

%计算累积概率

```
cumsump(1)=Pperpopulation(1);
```

```
for i=2:popsiz
```

```
    cumsump(i)=cumsump(i-1)+Pperpopulation(i);
```

```
end
```

```
cumsump=cumsump';
```

%子程序：判断遗传运算是否需要交叉或变异，函数名称存储为 IfCroIfMut.m

```
function pcc=IfCroIfMut(mutORcro);
```

```
test(1:100)=0;
```

```
l=round(100*mutORcro);
```

```
test(1:l)=1;
```

```
n=round(rand*99)+1;
```

```
pcc=test(n);
```

```
function[population]=makesureDifferent(population);
```

```
popsiz=size(population,1);
```

```
for i=1:popsiz
```

```
    individual=population(i,:);
```

```
    for i=2:7
```

```
        flag=individual(i)-individual(
```

%子程序：新种群变异操作，函数名称存储为 mutation.m

```
function snnew=mutation(snew,pmutation);
```

```
BitLength=size(snew,2);
```

```
snnew=snew;
```

```
pmm=IfCroIfMut(pmutation); %根据变异概率决定是否进行变异操作，1 则是，0 则否
```

```
if pmm==1
```

```
    chb=round(rand*(BitLength-1))+1; %在[1,BitLength]范围内随机产生一个变异位
```

```
    snnew(chb)=abs(snew(chb)-1);
```

```
end
```

%子程序：对于优化最大值或极大值函数问题，目标函数可以作为适应度函数

%函数名称存储为 targetfun.m

```
global Data
```

```

global Data_size
global flag
Data = xlsread('数据表格.xls','C2:BC1089');
Data_size =size( Data );

individual=[1 2 3 44 44 52 53]
individual=sort(individual);
%解决相同元素问题
pc=1;
while pc>0
    pc=-1;
    individual=sort(individual);
    for i=2:7
        flag=round(individual(i)-individual(i-1));
        if flag==0
            individual(i)=round(individual(i)+1);
            if round(individual(i))==54
                individual(i)=1;
                pc=1;
                individual=sort(individual);

                break
            end
        else
            individual(i)=individual(i);
        end
    end
end

end

individual=sort(individual);

Score = zeros( 1, Data_size(1)/2 );
for i = 1 : Data_size(1)/2
    Y = interp1( Data( 2*i-1 , individual(1, : ) ),Data( 2*i,
individual(1, : ) ),Data( 2*i-1 ,2:52 ),'spline');
    dert_Y = abs( Y(:, :) - Data( 2*i, 2:52 ) );

    for j = 1:( Data_size(2) - 2 )
        if dert_Y( j ) <= 0.01
            Score( 1, i ) = Score( 1, i ) + 2;
        elseif dert_Y( j ) <= 0.03
            Score( 1, i ) = Score( 1, i ) + 1.8;
        elseif dert_Y( j ) <= 0.05

```

```

        Score( 1, i ) = Score( 1, i ) + 1.2;
    elseif dert_Y( j ) <= 0.1
        Score( 1, i ) = Score( 1, i ) + 0.4;
    elseif dert_Y( j ) <= 0.15
        Score( 1, i ) = Score( 1, i ) + 0;
    else
        Score( 1, i ) = Score( 1, i ) - 20;
    end
end
end

```

```

Score( 1, i ) = Score( 1, i ) - 2;
if Score( 1, i ) < 0
    Score ( 1, i ) = 0;
end
end
y=mean(Score(1,:));

```

%子程序：对于优化最大值或极大值函数问题，目标函数可以作为适应度函数

%函数名称存储为 targetfun.m

function y=targetfun(s); %目标函数

global Data

global Data_size

global flag

```

s=sort(s);
%解决相同元素问题
for i=2:7
    for j=2:i
        flag=s(i)-s(j-1);
        if flag==0
            s(i)=s(i)+(i-j+1);
        end
    end
end
end
end

```

```

s=sort(s);

```

```

Score = zeros( 1, Data_size(1)/2 );

```

```

for i = 1 : Data_size(1)/2

```

```

    Y = interp1( Data( 2*i-1 , s(1, : ) ),Data( 2*i, s(1, : ) ),Data( 2*i-1 ,2:52 ),'spline');

```

```

    dert_Y = abs( Y(:, :) - Data( 2*i, 2:52 ) );

```

```

for j = 1:( Data_size(2) - 2 )

```

```

    if dert_Y( j ) <= 0.01

```

```

        Score( 1, i ) = Score( 1, i ) + 2;
    elseif dert_Y( j ) <= 0.03
        Score( 1, i ) = Score( 1, i ) + 1.8;
    elseif dert_Y( j ) <= 0.05
        Score( 1, i ) = Score( 1, i ) + 1.2;
    elseif dert_Y( j ) <= 0.1
        Score( 1, i ) = Score( 1, i ) + 0.4;
    elseif dert_Y( j ) <= 0.15
        Score( 1, i ) = Score( 1, i ) + 0;
    else
        Score( 1, i ) = Score( 1, i ) - 20;
    end
end

```

```

Score( 1, i ) = Score( 1, i ) - 2;
if Score( 1, i ) < 0
    Score ( 1, i ) = 0;
end
end
y=mean(Score(1,:));

```

%子程序：将 2 进制数转换为 10 进制数,函数名称存储为 transform2to10.m

```
function x=transform2to10(Population);
```

```
BitLength=size(Population,2);
```

```

for g=1:7
x(g)=round(Population(8*g));%BitLength=56
for i=1:7
    x(g)=round(x(g)+Population(8*g-i)*power(2,i));
end
end
x=round(sort(x));

```