

遗传算法: <https://www.zhihu.com/question/23293449>

模拟退火: <https://www.cnblogs.com/flashhu/p/8884132.html>

BP 神经网络: [https://blog.csdn.net/lyxleft/article/details/82840787?](https://blog.csdn.net/lyxleft/article/details/82840787?utm_medium=distribute.pc_relevant.none-task-blog-2_default_baidujs_baidulandingword~default-1.no_search_link&spm=1001.2101.3001.4242)

[utm\\_medium=distribute.pc\\_relevant.none-task-blog-2\\_default\\_baidujs\\_baidulandingword~default-1.no\\_search\\_link&spm=1001.2101.3001.4242](https://blog.csdn.net/lyxleft/article/details/82840787?utm_medium=distribute.pc_relevant.none-task-blog-2_default_baidujs_baidulandingword~default-1.no_search_link&spm=1001.2101.3001.4242)

[https://blog.csdn.net/weixin\\_40432828/article/details/82192709?](https://blog.csdn.net/weixin_40432828/article/details/82192709?utm_medium=distribute.pc_relevant.none-task-blog-2_default_baidujs_baidulandingword~default-5.no_search_link&spm=1001.2101.3001.4242)

[utm\\_medium=distribute.pc\\_relevant.none-task-blog-2\\_default\\_baidujs\\_baidulandingword~default-5.no\\_search\\_link&spm=1001.2101.3001.4242](https://blog.csdn.net/weixin_40432828/article/details/82192709?utm_medium=distribute.pc_relevant.none-task-blog-2_default_baidujs_baidulandingword~default-5.no_search_link&spm=1001.2101.3001.4242)

蚁群算法: <https://blog.csdn.net/u010425776/article/details/79517301>

神经网络编程入门: <https://www.cnblogs.com/heaad/archive/2011/03/07/1976443.html>

问题一

```
d285 = xlsread('3-1.xlsx')
```

```
d313 = xlsread('3-2.xlsx')
```

```
max = xlsread('附件四: 354 个操作变量信息.xlsx',2,'B1:B354')
```

```
min = xlsread('附件四: 354 个操作变量信息.xlsx',2,'A1:A354')
```

```
cmax = max.'
```

```
cmin = min.'
```

```
[row285,col285] = size(d285)
```

```
[row313,col313] = size(d313)
```

```
d285_maxmin = zeros(row285,col285)
```

```
for j = 1:col285
```

```
for i = 1:row285
```

```
if D285(i,j)>cmax(j) || D285(i,j)<cmin(j)
```

```
D285_maxmin(i,j) = 0;
```

```
else
```

```
D285_maxmin(i,j) = D285(i,j);
```

```
end
```

```
end
```

```
end
```

```
d313_maxmin = zeros(row313,col313)
```

```
% 对313操作变量按照最大最小幅度处理, 超过的数据赋值0
```

```
for j = 1:col313
```

```
for i = 1:row313
```

```
if D313(i,j)>cmax(j) || D313(i,j)<cmin(j)
```

```

D313_maxmin(i,j) = 0;
else
D313_maxmin(i,j) =D313(i,j);
end
end
end

zeros_num285_2 = zeros(row285,1)
for i = 1 : row285
zeros_num285_2(i) = length(find(d285_maxmin(i,:)==0))
end

zeros_num313_2 = zeros(row313,1)
for i = 1 : row313
zeros_num313_2(i) = length(find(d313_maxmin(i,:)==0))
end

%% 3、拉依达准则去除异常值
LD285 = zeros(row285,col285)
D285_maxmin_mean = mean(D285_maxmin); %计算各列算数平均值
D285_vi = D285_maxmin - D285_maxmin_mean;
D285_vi_pingfa = D285_vi.* D285_vi;
sum_D285_vi_pingfa = sum(D285_vi_pingfa);
dred285 =sqrt((sum_D285_vi_pingfa /(row285-1))); for j =1:col285
for i =1:row285
if abs(D285_vi(i,j))>3*dred285(j)
LD285(i,j) = 0;
else
LD285(i,j)=d285_maxmin(i,j);
end
end
end

LD313 = zeros(row313,col313);
D313_maxmin_mean = mean(D313_maxmin); %计算各列算数平均值
D313_vi = D313_maxmin - D313_maxmin_mean;
D313_vi_pingfa = D313_vi.* D313_vi;
sum_D313_vi_pingfa = sum(D313_vi_pingfa);
dred313 =sqrt((sum_D313_vi_pingfa /(row313-1))); for j =1:col313
for i =1:row313

```

```
if abs(D313_vi(i,j))>3*dred313(j)
```

```
LD313(i,j) = 0;
```

```
else
```

```
LD313(i,j)=D313_maxmin(i,j);
```

```
end
```

```
end
```

```
end
```

```
zeros_num285_3 = zeros(row285,1);
```

```
for i = 1:row285
```

```
zeros_num285_3(i) = length(find(LD285(i,:)==0)) ;
```

```
end
```

```
zeros_num313_3 = zeros(row313,1);
```

```
for i = 1:row313
```

```
zeros_num313_3(i) = length(find(LD313(i,:)==0)) ;
```

```
end
```

```
%% 4、找到每行大于 0 个数大于 20 的行号并剔除
```

```
hanghao285 = find(zeros_num285_3>20);
```

```
[row_hanghao285,col_hanghao285] = size(hanghao285);
```

```
LD285(hanghao285,:) = [];
```

```
quling_LD285 = LD285;
```

```
hanghao313 = find(zeros_num313_3>20);
```

```
[row_hanghao313,col_hanghao313] = size(hanghao313);
```

```
LD313(hanghao313,:) = [];
```

```
quling_LD313 = LD313;
```

```
%% 5、数据补齐
```

```
[row284_last,col285_last] = size(quling_LD285); [row313_last,col313_last] = size(quling_LD313);
```

```
twohours_mean285 = mean(quling_LD285); twohours_mean313 = mean(quling_LD313);
```

```
last_285 = zeros(row284_last,col285_last); last_313 = zeros(row313_last,col313_last);
```

```
for i = 1:row284_last
```

```
for j = 1:col285_last
```

```
if quling_LD285(i,j) == 0
```

```
last_285(i,j) =twohours_mean285 (1,j);
```

```
else
```

```
last_285(i,j)=quling_LD285(i,j);
```

```
end
```

```

end
end

for i = 1:row313_last
for j = 1:col313_last
if quling_LD313(i,j) == 0
last_313(i,j)=twohours_mean313 (1,j);
else
last_313(i,j)=quling_LD313(i,j);
end
end
end
end

```

%% 5、求平均值

```

mean_last285 = mean(last_285);
mean_last313 = mean(last_313);

```

问题二

```

clc
clear all

```

```

[num,txt,row]=xlsread('附件一： 325个样本数据.xlsx') ;
x = num(2:end,2:end)';
index_num = size(x,1)
column_num = size(x,2);

```

% 1、数据均值化处理

```

x_mean = mean(x,2);
for i = 1:index_num
x(i,:) = x(i,+)/x_mean(i,1);
end

```

%2、提取参考列和比较列数据

```

ck = x(1,:);
cp=x(2:end,:);
cp_index_num = size(cp,1);
y = cp;
x = ck;
y_row = size(y,1);%%计算矩阵 y 的行数
y_col =size(y,2);%%计算矩阵 y 的列数
x_col = size(x,2);%%计算 x 的列数

```

```

if y_col ~= x_col
error(message('MATLAB:greyrelation:wrong in input data'));
end

temp_y = y;%绝对关联度中比较序列中的数据处理后的矩阵temp_x = x;%x 数据处理后的矩阵

for i =1:x_col
temp = x(i)-x(1);
temp_x(i)=temp;
end

for i =1:y_row
for j=1:y_col
temp = y(i,j) - y(i,1);
temp_y(i,j)=temp;
end
end

%处理过程
%temp_x;
%temp_y;
s0 = abs(sum(temp_x)-0.5temp_x(x_col));
abs_xy = [];
for i=1:y_row
si = abs(sum(temp_y(i,:))-0.5temp_y(i,y_col)); si_s0 = abs(si-s0);
abs_xy(i,1) =(1+s0+si)/(1+s0+si+si_s0);
end

%下面开始计算相关关联度
temp_y2 = y;
temp_x2 = x;
for i =1:x_col
temp = x(i)/x(1);
temp_x2(i)=temp-1;
end

for i =1:y_row
for j=1:y_col
temp = y(i,j) / y(i,1);
temp_y2(i,j)=temp-1;
end

```

```

end
s02 = abs(sum(temp_x2)-0.5temp_x2(x_col)); rela_xy=[];
for i=1:y_row
    si2 = abs(sum(temp_y2(i,:))-0.5temp_y2(i,y_col));
    si2_s02 = abs(si2-s02);
    rela_xy(i,1) =(1+s02+si2)/(1+s02+si2+si2_s02)
end

%下面计算综合关联度
com_xy = 0.5*abs_xy +(1-0.5)*rela_xy;%返回的是综合关联度

```

### 问题三

```

clc
clear
data = xlsread('附件一： 325个样本数据 - 副本.xlsx')
input_train = data(1:300,2:end)
output_train = data(1:300,1:2)

input_test = data(301:end,2:end)
output_test = data(301:end,1:2)

input_train = input_train.'
output_train = output_train.'
input_test = input_test.'
output_test = output_test.'

%训练数据归一化
[inputn,inputps] = mapminmax(input_train);
[outputn,outputps] = mapminmax(output_train);

net = newff(inputn,outputn,90);
%参数设置
net.trainParam.epochs=100;%迭代次数
net.trainParam.lr=0.4;%学习率
net.trainParam.goal=0.0000000001;%收敛目标
%神经网络训练
net = train(net,inputn,outputn);
%训练数据归一化
inputn_test = mapminmax('apply',input_test,inputps);
%神经网络测试输出

```

```

an = sim(net,inputn_test);
BPoutput = mapminmax('reverse',an,outputps);
%数据可视化
figure(1)
plot(BPoutput(1,:), 'g') %红
hold on
plot(output_test(1,:), 'k. ');
hold on
plot(BPoutput(2,:), 'r') %
hold on
plot(output_test(2,:), 'b. ');
legend('模拟值(含硫量)', '原始值 (含硫量)', '模拟值 (辛烷值)', '原始值 (辛烷值)')
err = abs(BPoutput - output_test);
err_mean = mean(err);
figure(2)
plot(err_mean, '-*')
title('测试误差')
ylabel('平均误差')
xlabel('样本')

```

#### 问题四

```

clc
clear
close all
x1 = 89.8; %设定原材料辛烷值
x2 = 56.10; %饱和烃, v% (烷烃+环烷烃)
x_index = 173; %优化数据编码
y0 = 89.22; %原始数据产成品辛烷值
[num_input, txt_input, raw_input] = xlsread('input'); [num_output, txt_output, raw_output] = xlsread('output')
;
input = num_input;
output = num_output;
data_train_input = input(1:300,:);
data_train_output = output(1:300,:);
input_train = data_train_input'; %30300
output_train = data_train_output'; %2300

%训练数据归一化
[inputn, inputps] = mapminmax(input_train);

```

```

[outputn,outputps] = mapminmax(output_train);
%参数设置
net = newff(inputn,outputn,90);
net.trainParam.epochs=100;%迭代次数
net.trainParam.lr=0.4;%学习率
net.trainParam.goal=0.0000000001;%收敛目标

%神经网络训练
net = train(net,inputn,outputn);
%对 x1, x2 标准化数据获取
x11 = inputn(1,x_index);
x22 = inputn(2,x_index);

%% 遗传参数设置
popsize = 50; %种群大小
pc = 0.7; %交叉率
pm = 0.09; %变异率
Iteration =1000; %最大迭代次数
nodes = 28;
r_lost_mean = 1.2; %辛烷值损失值均值

%% 初始化总群
initPop = zeros(nodes,popsize);
for j = 1:popsize
    r = rand(nodes+2,popsize);
    for i = 1:nodes+2
        initPop(i,j) = -1+2*r(i,j);
    end
    r=[];
end

initPop(1,:)=x11;
initPop(2,:)=x22;
trace =zeros(Iteration,3);%第一列存迭代次数, 第二列存最小误差, 第三列存平均误差
NewPop = zeros(nodes+2,popsize);%选择后的种群
for gen = 1:Iteration
    %% 选择操作
    % 计算适应度
    an = sim(net,initPop);
    an_output = mapminmax('reverse',an,outputps); fitness = zeros(1,popsize);

```



```

for j = 1:popsiz
if an_output(1,j)<0||(x1-an_output(2,j)-r_lost_mean)/r_lost_mean>0.3||(x1- an_output(2,j))
<0||an_output(1,j)>5
fitness(j) = 0.000001;
else
fitness(j) = 1/(x1-an_output(2,j));
end
end
[mmin index_err] = max(fitness);
%计算选择概率
pz = fitness./sum(fitness);
%计算概率累计
qz = sum(pz);
%执行选择
index = zeros(1,popsiz);
for i = 1:popsiz
pick = rand;
while pick == 0
pick = rand;
end
pick = pick - pz(index_err);
if pick<0
index(1,i) = index_err;
else
index(1,i) = ceil(popsiz*rand);
end
end
NewPop = initPop(:,index);
%% 交叉操作
for j =1:popsiz
pick = rand(1,2);
while prod(pick) == 0
pick = rand(1,2);
end
index = ceil(pick * popsiz);
%交叉率是否决定交叉
pick = rand;
while pick == 0
pick = rand;

```

```

end
if pick > pc
continue
end
flag = 0;
pick1 = rand;
pick2 = rand;
pos1 = ceil(pick1 * (nodes+2));
pos2 = ceil(pick2 * (nodes+2));
while pos1 == pos2
pick1 = rand;
pick2 = rand;
pos1 = ceil(pick1 * nodes);
pos2 = ceil(pick2 * nodes);
pos1 = min(pos1,pos2);
pos2 = max(pos1,pos2);
v1 = NewPop(pos1:pos2,index(1));
v2 = NewPop(pos1:pos2,index(2));
NewPop(pos1:pos2,index(1)) = v2;
NewPop(pos1:pos2,index(2)) = v1;
end
end

```

%% 变异操作

```

for j = 1:popsiz
r1 = rand(nodes+2,popsiz);
pick = rand;
if pick > pm
continue;
end
%变异位置
pick = rand;
while pick == 0
pick = rand;
end
index = ceil(pick*(nodes+2));
pickj = rand;
while pickj == 0
pickj = rand;

```

```

end
j=ceil(pickj*(nodes+2));
if index1
NewPop(index,j) = x11;
elseif index2
NewPop(index,j) = x22;
else
NewPop(index,j) = -1+2*r1(index,j);
end
end

%计算误差
bn = sim(net,NewPop);
bn_output = mapminmax('reverse',bn,outputps);
err1 =zeros(1,popsize);
for i =1:popsize
err1(1,i) = abs(x1-bn_output(2,i));
end

%计算每代平均误差
aveErr = sum(err1)/popsize;
[minErr,bestIndex] = min(err1);
bestinputps = NewPop(:,bestIndex);
trace (gen,1) = gen;
trace (gen,2) = minErr;
trace (gen,3) = aveErr;
initPop = NewPop;
end

x = trace(:,1);
minerr = trace(:,2);
avgerr = trace(:,3);
figure
plot(x,minerr,'r--',x,avgerr,'b-');
xlabel('Iterations');
ylabel('ERR');
legend('minerr','avgerr');
grid;
an = sim(net,bestinputps);

```

```
BPoutput = mapminmax('reverse',an,outputps)
```

```
bsetChrombsetChrom_rever= mapminmax('reverse',bestinputps,inputps);
```