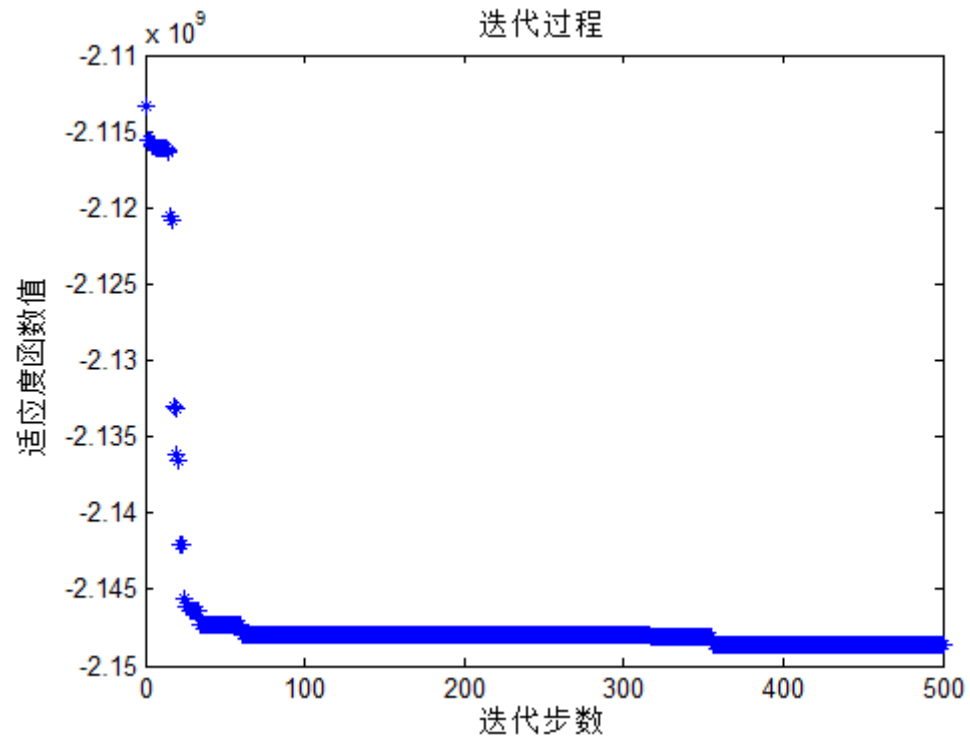


第一问 特别注意啊这还不是全局最优 我还算了一次

Elapsed time is 73.420506 seconds.

这句话的意识是程序跑了一圈 500 步，花了 73.420506 秒 还是可以需要展示出来



这是迭代收敛的过程，我通过比较发现 500 步和 1000 步的结果差不多，估计应该是全局最优了。

但是这是适应度函数的值还不是发电量。别写错了！

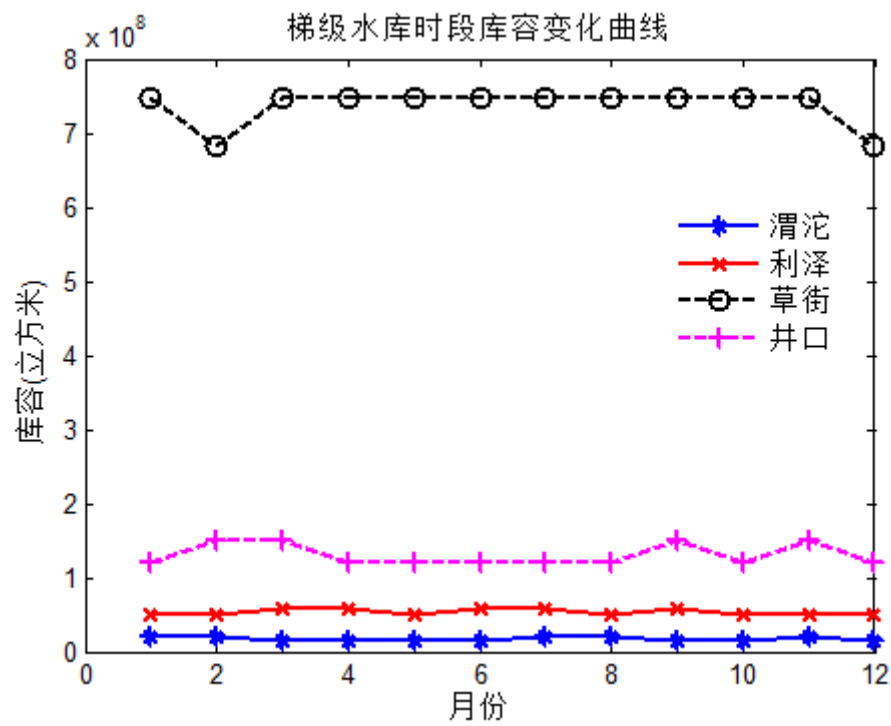
做个表格把这些参数展示一下!!!!!!!!!!!!!!

年发电量：=2148667234.7011

最优调度方案:

x =

19300000	49220000	746670000	120950000
19300000	49220000	683330000	149520000
13900000	57730000	746670000	149520000
13900000	57730000	746670000	120950000
13900000	49220000	746670000	120950000
13900000	57730000	746670000	120950000
19300000	57730000	746670000	120950000
19300000	49220000	746670000	120950000
13900000	57730000	746670000	149520000
13900000	49220000	746670000	120950000
19300000	49220000	746670000	149520000
13900000	49220000	683330000	120950000



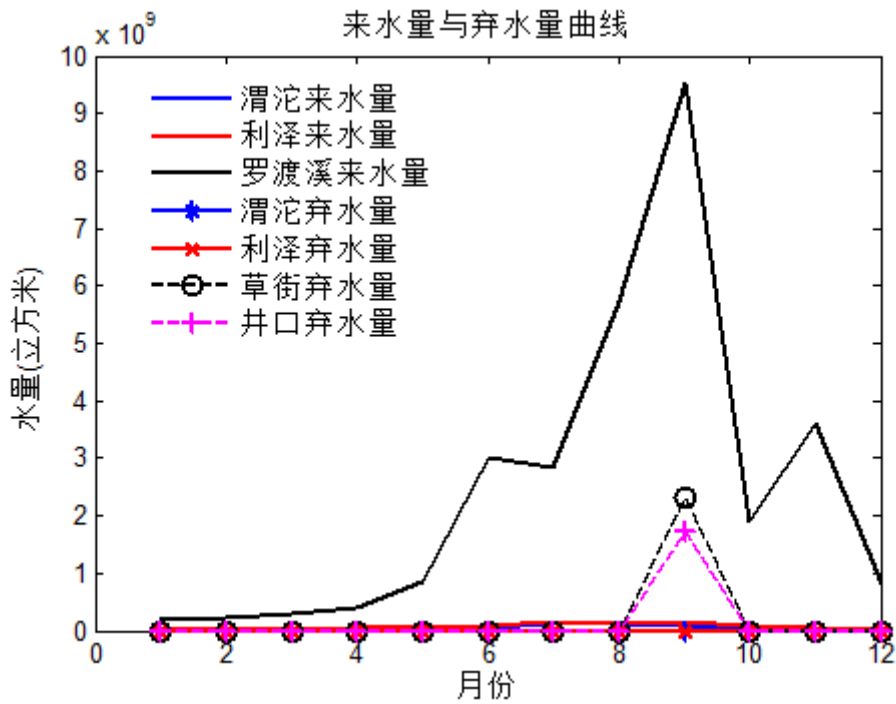
草街处于汇流的地段，来水量充足，库容曲线变化较小

弃水情况:

discharge =

1.0e+09 *

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	2.3040	1.7184
0	0	0	0
0	0	0	0
0	0	0	0



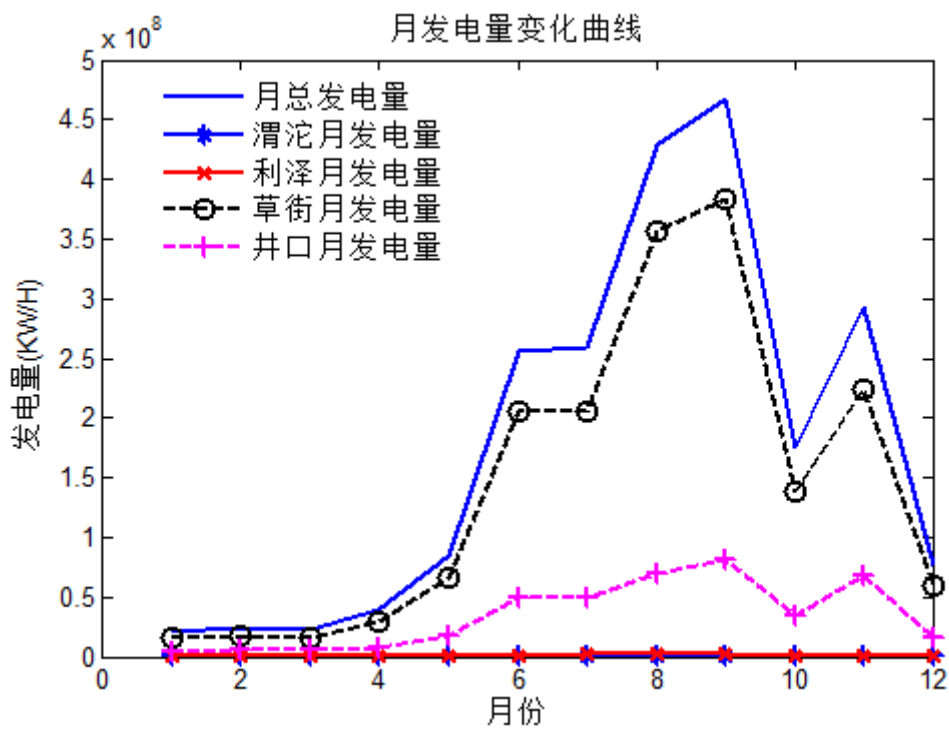
还是6月份到10月份间，渭沱利泽有一定的弃水，而草街和井口仅仅在来水量最多的9月份有集中的弃水。

各电站月出力:

Es =

1.0e+08 *

0.000425325230127	0.004728007783399	0.159505108592378	0.045804067870085
0.000756298187577	0.003365447292199	0.184137465729352	0.056455613070555
0.000444932729873	0.002204078822340	0.164306335846192	0.053059050069678
0.000516249611259	0.006972205691228	0.300896030902976	0.081910438063778
0.001003373678073	0.011973253015293	0.660882020397799	0.168265732449417
0.001409328340366	0.014622101199934	2.052418688401477	0.494134742308967
0.006343562838147	0.031159841764169	2.060457445536141	0.495793980435168
0.006225753893840	0.024074746073821	3.563419360343882	0.689866115322698
0.003189898892318	0.029176433401730	3.826505227908057	0.821581840206949
0.001468713709721	0.016763664853483	1.384412043268529	0.349778234207382
0.001105500554297	0.007662278183252	2.248028769985681	0.679392923040775
0.001229206256526	0.005150117782876	0.599020087858666	0.164670705408402



库容基本没变，都是正常水位，即保持在高水头运行状态。

这里可以看到 平时都没有弃水只有在来水比较多的9月份才有弃水。。。。。。。。。

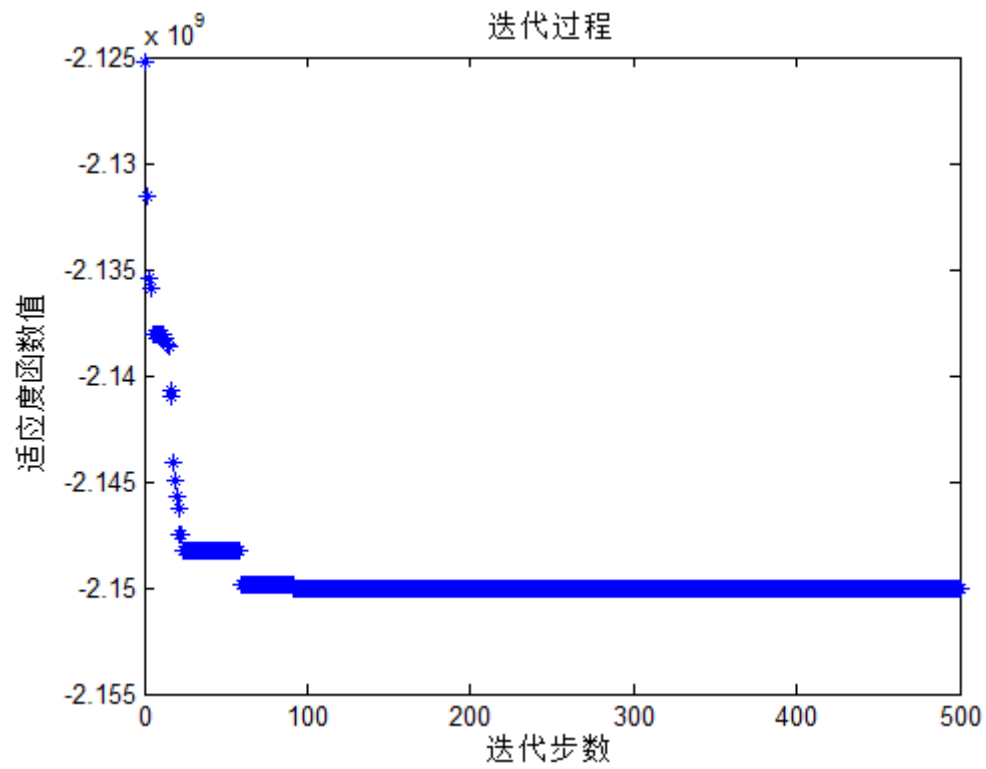
以上就是我的分析该怎么写结合华中科技大学的博士论文。。。

第二问：我们也不管他怎么样就假定渭沱电站库容上限被提高了 10%然后再进行计算看看结果如何

以下是计算结果：

Elapsed time is 247.667838 seconds.

解的收敛过程：

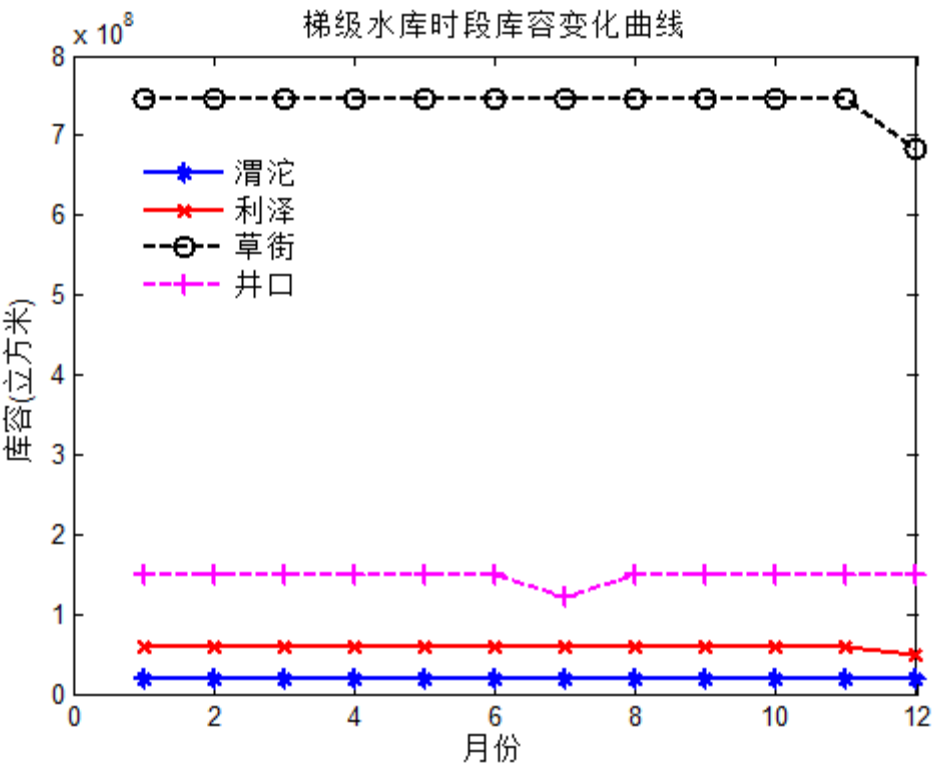


这个算法有时候算的快有时候算的慢，但是收敛的速度还是可以的。

年发电量：=2150089557.311
最优调度方案：

x =

19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	120950000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	49220000	683330000	149520000

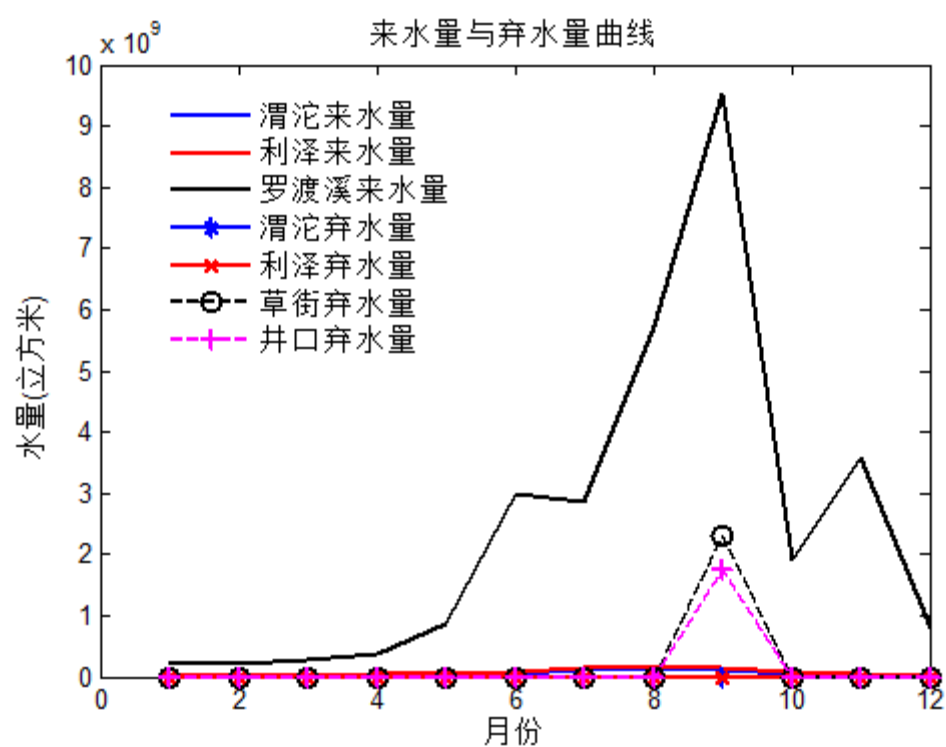


弃水情况:

discharge =

1.0e+09 *

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	2.3071	1.7500
0	0	0	0
0	0	0	0
0	0	0	0

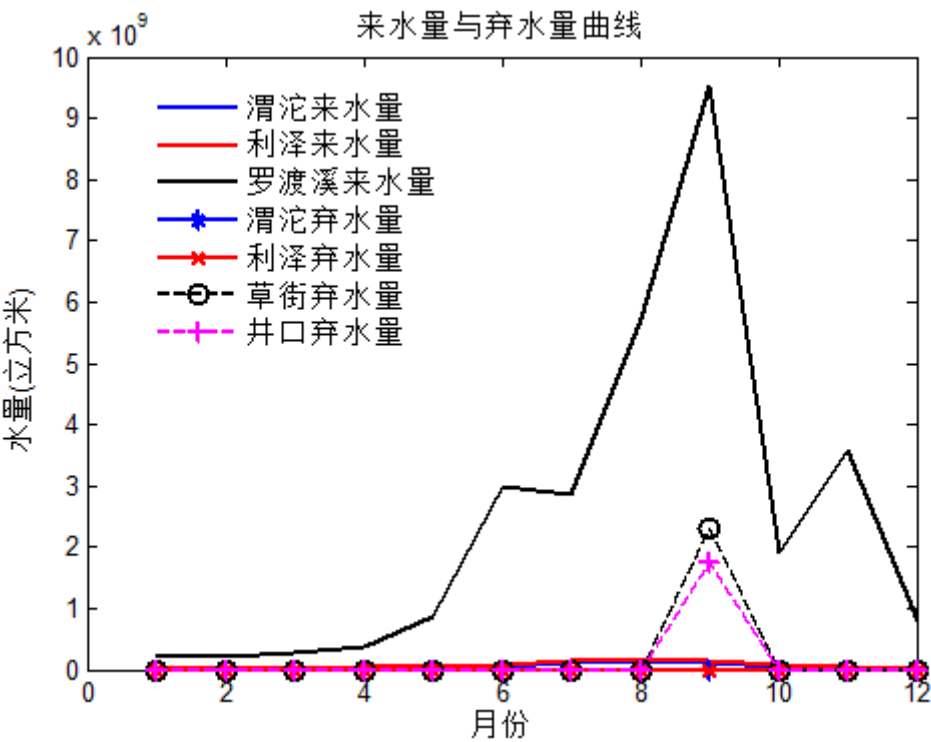


各电站月出力:

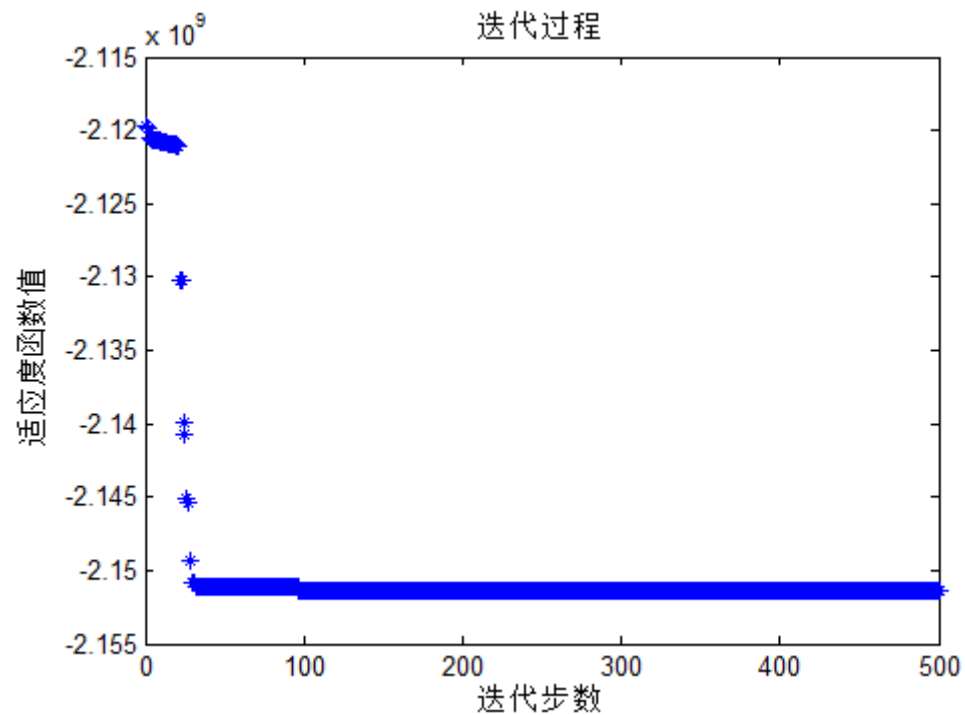
Es =

1.0e+08 *

0.000425325230127	0.003871648525637	0.145927812955557	0.047126800096968
0.000483573631631	0.003318562281596	0.152352449404383	0.049200669484287
0.000511741053547	0.003854888384590	0.207255939028936	0.066917993506797
0.000934203313767	0.006972205691228	0.285655380159648	0.092195641486774
0.001816178114608	0.012334821123418	0.621870601592343	0.200090852815648
0.002551875145211	0.016272359386799	1.949720096767172	0.603102804638934
0.006631997703121	0.031159841764169	2.063752002934605	0.500049787977836
0.006225753893840	0.026472588940190	3.353408956989007	0.895401310025856
0.005503526821206	0.030825100702780	3.826505227908057	0.821581840206949
0.002659566721389	0.017930942367829	1.308020617951655	0.414952547078230
0.001398755118728	0.008949761923380	2.250997289026973	0.684252113173677
0.001262899448200	0.006824188665743	0.569068915693505	0.192295616253309



第三问：首先展示解得收敛情况然别人知道结果是可信的



收敛的很快，算法的效果还不错。

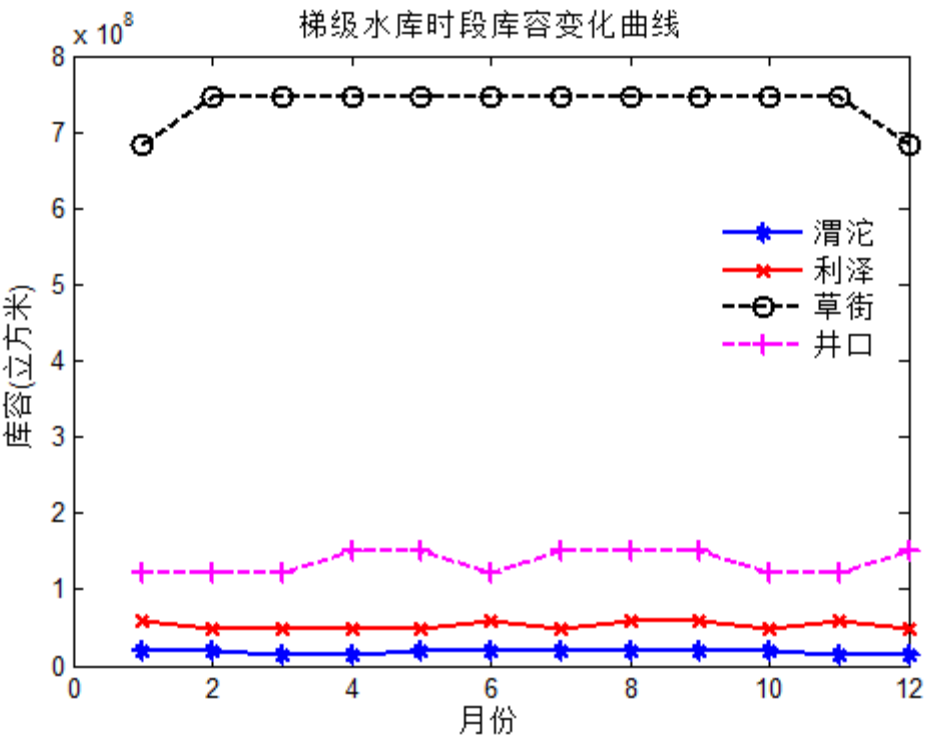
第三问中要求提高水轮机的功率，那么根据公式就相当于要提高水轮机的水头和流量，先将水轮机的流量提高 5%和水轮机水头限制的上限提高 5%进行再度优化比较和以前的结果。

以下是数值结果（需要列表） 可以看到年总发电量从 2148667234.7011 提高到了 2151329105.2073

年发电量：=2151329105.2073
最优调度方案：

x =

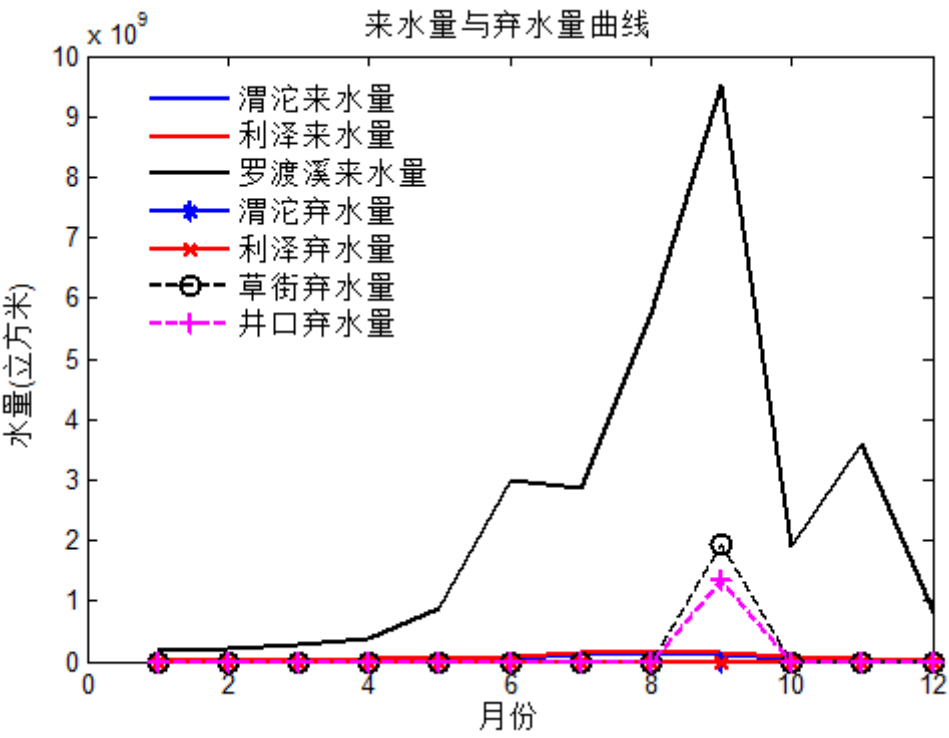
19300000	57730000	683330000	120950000
19300000	49220000	746670000	120950000
13900000	49220000	746670000	120950000
13900000	49220000	746670000	149520000
19300000	49220000	746670000	149520000
19300000	57730000	746670000	120950000
19300000	49220000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	57730000	746670000	149520000
19300000	49220000	746670000	120950000
13900000	57730000	746670000	120950000
13900000	49220000	683330000	149520000



弃水情况:
discharge =

1.0e+09 *

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	1.9332	1.3483
0	0	0	0
0	0	0	0
0	0	0	0

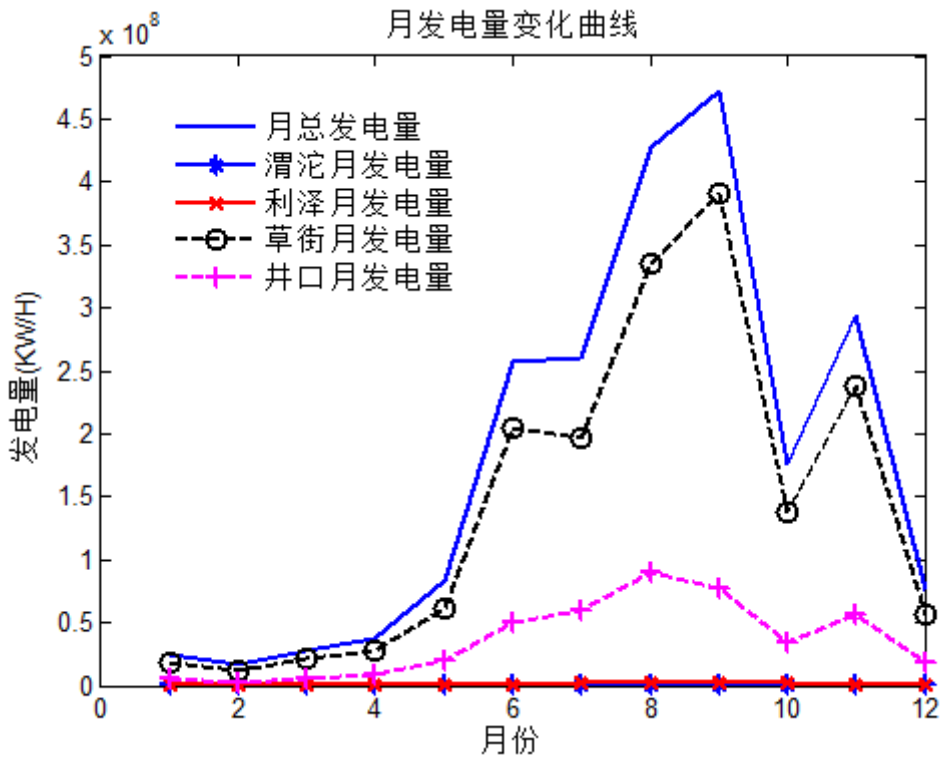


各电站月出力:

Es =

1.0e+08 *

0.000665197313495	0.004483292918100	0.187985958685658	0.055347937810962
0.000483573631631	0.004254496889746	0.123130815234509	0.031521440163853
0.000444932729873	0.003300354189404	0.221985566196688	0.056803439839207
0.000516249611259	0.005969222103413	0.285655380159648	0.086251194913985
0.001523067091890	0.010560311698822	0.618413344694025	0.198987175387677
0.002551875145211	0.014622101199934	2.052418688401477	0.497728160471030
0.006631997703121	0.028086245099724	1.960405825027101	0.601496295373866
0.006225753893840	0.024823338788933	3.350111940627799	0.896052222639581
0.005503526821206	0.030825100702780	3.908976687195532	0.770366245072563
0.002659566721389	0.016763664853483	1.384412043268529	0.349778234207382
0.000934765887357	0.007299088905378	2.375529377642084	0.557567438689111
0.000901653031962	0.006824188665743	0.569068915693505	0.186443159079302



下面讲解我的算法中的一些参数

1. 这个函数是整个算法的核心也就是混沌混合 PSO 算法的程序

其中的参数是这样取取得

由于问题维数比较高，故粒子数目选择 200 个 N，个体学习因子 c1 取得是 1，群体学习因子取得也是 1，自变量搜索范围就选的库容的变化范围，最大迭代次数为 500 次，其实到 300 次就已经较好的收敛了，惩罚系数 r 为 10^8 这里的权重是变化权重按下列公式这样选取：

$$\omega(k) = \omega_{\text{start}} - (\omega_{\text{start}} - \omega_{\text{end}}) \left(\frac{k}{T_{\text{max}}} \right)^2$$

w 为权重 起始权重，终止权重， Tmax 最大迭代次数 k 当前迭代次数

Maxc 为最大混沌搜索迭代次数选择为 100 次

```
function xm=CLSPSO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 混沌 PSO 联合优化算法
%待优化函数：target_maxf
%粒子数目：N
%惯性权重：w
%学习因子 1：c1
%学习因子 2：c2
%自变量搜索域的最大值：xmax
%自变量搜索域的最小值：xmin
%最大迭代次数:M
%混沌搜索的最大步数:MaxC
%问题的维数：D
%惩罚系数 r
%目标函数取最小值时的自变量值：x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 定义算法参数
tic;
N=200;
D=48;
w=0.8;
c1=1;
c2=1;
M=500;
MaxC=100;
r=1e8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 调节显示长度
format long;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 初始化变量
x=zeros(N,D); %记录种群位置
v=x;          %记录种群速度
p=zeros(1,N); %记录粒子局部最佳适应度
y=x;          %记录粒子局部最佳位置
Iteration=0;  %记录迭代步数
```

%% 随机初始化位置

for i=1:N

for j=1:4:D

x(i,j)= unifrnd(0.1390e8,0.193e8);

x(i,j+1)=unifrnd(0.4922e8,0.5773e8);

x(i,j+2)=unifrnd(6.8333e8,7.4667e8);

x(i,j+3)=unifrnd(1.2095e8,1.4952e8);

end

end

%% 随机初始化速度

for i=1:N

for j=1:4:D

v(i,j)= unifrnd(0.0090e8,0.0093e8);

v(i,j+1)=unifrnd(0.004922e8,0.00773e8);

v(i,j+2)=unifrnd(0.08333e8,0.04667e8);

v(i,j+3)=unifrnd(0.02095e8,0.04952e8);

end

end

%% 计算初始值

for i=1:N

p(i)=fitness(x(i,:),r);

y(i,:)=x(i,:);

end

pg=x(N,:); %pg 为全局最优对应的粒子

for i=1:(N-1)

if fitness(x(i,:),r)<fitness(pg,r)

pg=x(i,:);

end

end

%% 设置速度限制 20%

vmax=zeros(1,D);

vmin=zeros(1,D);

for j=1:4:D

vmin(j)=0.2* (0.1390e8-0.193e8);

vmin(j+1)=0.2*(0.4922e8-0.5773e8);

vmin(j+2)=0.2*(6.8333e8-7.4667e8);

vmin(j+3)=0.2*(1.2095e8-1.4952e8);

vmax(j)=-vmin(j);

vmax(j+1)=-vmin(j+1);

vmax(j+2)=-vmin(j+2);

vmax(j+3)=-vmin(j+3);

end

%% 设置位置限制

lb=zeros(1,D);

```

ub=zeros(1,D);
for j=1:4:D
    lb(j)= 0.1390e8;
    lb(j+1)=0.4922e8;
    lb(j+2)=6.8333e8;
    lb(j+3)=1.2095e8;
    ub(j)= 0.1930e8;
    ub(j+1)=0.5773e8;
    ub(j+2)=7.4667e8;
    ub(j+3)=1.4952e8;
end
xmax=ub;
xmin=lb;
figure('Color',[1 1 1]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 主循环
for t=1:M
    for i=1:N
        v(i,:)=w*v(i,:)+c1*rand*(y(i,:)-x(i,:))+c2*rand*(pg-x(i,:)); %更新速度
        for k=1:D
            if v(i,k)>vmax(k)
                v(i,k)=vmax(k); %速度限制
            end
            if v(i,k)<vmin(k)
                v(i,k)=vmin(k);
            end
        end
        x(i,:)=x(i,:)+v(i,:); %更新位置
        for k=1:D
            if x(i,k)>ub(k)
                x(i,k)=ub(k); %位置限制
            end
            if x(i,k)<lb(k)
                x(i,k)=lb(k);
            end
        end
        fv(i)=fitness(x(i,:),r); %更新适应度
    end
    [sort_fv,index]=sort(fv);
    Nbest=floor(N*0.2); %保留群体中 20%的最佳粒子
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for n=1:Nbest %对群体中 20%的最佳粒子进行混沌搜索
        tmpx=x(index(n),:);
        for k=1:MaxC %混沌搜索的最大步数
            for dim=1:D %混沌搜索的迭代公式

```



```

        cx(dim)=(tmpx(1,dim)-xmin(dim))/(tmpx(1,dim)-xmax(dim));
        cx(dim)=4*cx(dim)*(1-cx(dim));
        tmpx(1,dim)=tmpx(1,dim)+cx(dim)*(xmax(dim)-xmin(dim));
    end
    fcs=fitness(tmpx,r);
    if fcs<sort_fv(n)                %对混沌搜索后的决策变量值进行评估
        x(index(n,:)=tmpx;
        break;
    end
end
x(index(n,:)=tmpx;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:N
    for k=1:D
        if x(i,k)>ub(k)
            x(i,k)=ub(k);          %位置限制
        end
        if x(i,k)<lb(k)
            x(i,k)=lb(k);
        end
    end
end
r=rand();
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for s=1:D                            %收缩搜索区域
    xmin(s)=max(xmin(s),pg(s)-r*(xmax(s)-xmin(s)));
    xmax(s)=min(xmax(s),pg(s)+r*(xmax(s)-xmin(s)));
end
x(1:Nbest,:)=x(index(1:Nbest,:);
for i=(Nbest+1):N                    %随机产生剩余的 80%微粒
    for j=1:D
        x(i,j)=xmin(j)+rand*(xmax(j)-xmin(j));    %随机初始化位置
        v(i,j)=0.2*rand*(xmax(j)-xmin(j));        %随机初始化速度
    end
end
for i=1:N
    for k=1:D
        if v(i,k)>vmax(k)
            v(i,k)=vmax(k);                %速度限制
        end
        if v(i,k)<vmin(k)

```

```

        v(i,k)=vmin(k);
    end
end

for k=1:D
    if x(i,k)>ub(k)
        x(i,k)=ub(k);    %位置限制
    end
    if x(i,k)<lb(k)
        x(i,k)=lb(k);
    end
end
end
end
Pbest(t)=fitness(pg,r);
for i=1:N
    if fitness(x(i,:),r)<p(i)
        p(i)=fitness(x(i,:),r);
        y(i,:)=x(i,:);
    end
    if p(i)<fitness(pg,r)
        pg=y(i,:);
    end
end
Iteration=Iteration+1;    %算法步数计数器自增
plot(Iteration,fitness(pg,r),'b*');
hold on;
w=0.9-(0.9-0.4)*(Iteration/M)^2;    %调节权重
end
hold off;
xm=pg;
save x_best xm
toc;

```

2. 目标函数和惩罚函数:

构造过于复杂不好讲的

```
function [result,violent,n,discharge,Em,Es]= target_maxf(x,r)
%四水电站混联 联合调度目标函数与罚函数
%默认 x 向量的编码方式为:
%x(j)代表库容 其中 4 个电站一个时间段为一个单元 共 12 个单元 48 个元素 一行排列
%函数返回三个数据分别为 目标函数值 result 约束违反程度 violent 与弃水量 discharge
%-----
%-----基本数据-----
%%%%%%%%%%%% 每月来水量, 丰水年情况
lize_month=[7.45161290322581,6.38709677419355,7.41935483870968,...
    13.4193548387097,23.7419354838710,31.3225806451613,...
    60.0000000000000,50.9677419354839,59.3548387096774,...
    34.5161290322581,17.2258064516129,9.77419354838710];%利泽月来水量
weituo_month=[3.96774193548387,3.32258064516129,3.51612903225807,...
    6.41935483870968,12.4838709677419,17.5483870967742,...
    45.8387096774194,43.0000000000000,37.9677419354839,...
    18.2903225806452,9.61290322580645,5.54838709677419];%渭沱月来水量
luoduxi_month=[73.8580645161291,78.2322580645161,108.719354838710,...
    145.135483870968,323.912903225806,1119.05161290323,...
    1065.83870967742,2140.51612903226,3556.06451612903,...
    714.161290322581,1340.83870967742,303.761290322581];%罗渡溪月来水量
%%%%%%%%%%%%每月秒数
t=31*24*60*60;
% 各电站最大发电引用流量
qm1=2*200; % 渭沱两台机组
qm2=4*388.9; % 利泽 4 台机组
qm3=4*698; % 草街 4 台机组
qm4=6*500; % 井口 4 台机组
% 边界条件水库初始库容
v01=1650e4;
v02=5773e4;
v03=74667e4;
v04=14952e4;
%-----
z=zeros(size(x)); % 记录库容对应的水位
discharge=z; % 记录个水库的弃水量
q=z; % 记录水库发电引用流量
h=z; % 记录上下游水位差
H=z; % 记录水轮机水头
Em=zeros(1,12); % 记录月总发电量
Es=z; %记录各电站每月发电量
    for j=1:4:48 %%%%%%%%%%% 库容推求水位 利用 cubic curve fitting
```

%%
渭沱上游水位对应的库容

z(j)=0.0018305874521551467135754842630035*x(j)/1e4+...
202.48285904476608720869990065694;

%%
利泽上游对应的水位

z(j+1)=0.1988639984e-2*x(j+1)/1e4-0.6896937851e-7*(x(j+1)/1e4)^2+...
0.2023282902e3+0.9590483713e-12*(x(j+1)/1e4)^3;

%%
草街上游对应水位

z(j+2)=0.3362104882e1*(x(j+2)/1e4)^3/0.1000000000000000e16+(-0.1e1)*...
0.1890378905e1*(x(j+2)/1e4)^2/0.1000000000e10+0.4154782216e-3*...
(x(j+2)/1e4)+ 0.1819231930e3;

%%
井口上游对应的水位

z(j+3)=0.6152201768e1*(x(j+3)/1e4)^3/0.1000000000000000e16+(-0.1e1)*...
0.2928489463e1*(x(j+3)/1e4)^2/0.1000000000e10+0.5667546633e-3*...
(x(j+3)/1e4)+0.1696579477e3;

end

%% 计算弃水量

%计算第一个月的

discharge(1)=(v01 - x(1)) / t + weituo_month(1)-qm1)*t;

if discharge(1)<0

discharge(1)=0;

end

discharge(2)=(v02-x(2)) / t + lize_month(1)-qm2)*t;

if discharge(2)<0

discharge(2)=0;

end

discharge(3)=(v03 - x(3) + v01 - x(1) + weituo_month(1) *...

t + v02 - x(2) + lize_month(1) * t + luoduxi_month(1) * t)-qm3*t);

if discharge(3)<0

discharge(3)=0;

end

discharge(4)=(v04 - x(4) + v03 - x(3) + v01 - x(1) + weituo_month(1)...

* t + v02 - x(2) + lize_month(1) * t + luoduxi_month(1) * t)-qm4*t);

if discharge(4)<0

discharge(4)=0;

end

%计算 2 到 12 月

for j=5:4:48

discharge(j)=(x(j-4) - x(j)) / t + weituo_month((j-1)/4+1)-qm1)*t;

```

        if discharge(j)<0
            discharge(j)=0; % 渭沱弃水量
        end
        discharge(j+1)=((x(j-3)-x(j+1)) / t + lize_month((j-1)/4+1)-qm2)*t;
        if discharge(j+1)<0
            discharge(j+1)=0; % 利泽弃水量
        end
        discharge(j+2)=((x(j-2) - x(j+2) + x(j-4) - x(j) + weituo_month((j-1)/4+1) *...
        t + x(j-3) - x(j+1) + lize_month((j-1)/4+1) * t + luoduxi_month((j-1)/4+1) * t)-qm3*t);
        if discharge(j+2)<0
            discharge(j+2)=0; % 草街弃水量
        end
        discharge(j+3)=((x(j-1) - x(j+3) + x(j-2) - x(j+2) + x(j-4) - x(j) +
        weituo_month((j-1)/4+1)...
        * t + x(j-3) - x(j+1) + lize_month((j-1)/4+1) * t + luoduxi_month((j-1)/4+1) * t)-qm4*t);
        if discharge(j+3)<0
            discharge(j+3)=0; % 井口弃水量
        end
    end
    %%%%%%%%%%% 发电引用流量计算
    %计算第一个月的
    q(1)=(v01-x(1)-discharge(1))/t+weituo_month(1);
    if q(1)<0
        q(1)=0;
    end
    q(2)=(v02-x(2)-discharge(2))/t+lize_month(1);
    if q(2)<0
        q(2)=0;
    end
    q(3)=(v03-x(3)-discharge(3))/t+q(1)+discharge(1)/t+q(2)+discharge(2)/t+luoduxi_month(1);
    if q(3)<0
        q(3)=0;
    end
    q(4)=(v04-x(4)-discharge(4))/t+q(3)+discharge(3)/t;
    if q(4)<0
        q(4)=0;
    end
    %计算 2 到 12 月
    for j=5:4:48
        q(j)=(x(j-4)-x(j)-discharge(j))/t+weituo_month((j-1)/4+1);
        if q(j)<0
            q(j)=0; % 渭沱发电引用流量
        end
        q(j+1)=(x(j-3)-x(j+1)-discharge(j+1))/t+lize_month((j-1)/4+1);
    end

```

```

        if q(j+1)<0
            q(j+1)=0; % 利泽发电引用流量
        end
        q(j+2)=(x(j-2)-x(j+2)-discharge(j+2))/t+q(j+1)+discharge(j+1)/t+...
            q(j)+discharge(j)/t+luoduxi_month((j-1)/4+1);
        if q(j+2)<0
            q(j+2)=0; % 草街发电引用流量
        end
        q(j+3)=(x(j-1)-x(j+3)-discharge(j+3))/t+q(j+2)+discharge(j+2)/t;
        if q(j+3)<0
            q(j+3)=0; % 井口发电引用流量
        end
    end

    %%%%%%%%%%%%%% 水位差计算
    for j=1:4:48
        h(j)=z(j)-z(j+2); % 渭沱草街水位差
        h(j+1)=z(j+1)-z(j+2); % 利泽草街水位差
        h(j+2)=z(j+2)-z(j+3); % 草街井口水位差
        h(j+3)=z(j+3)-169; % 井口嘉陵江水位差
    end

    %%%%%%%%%%%%%% 水轮机水头计算
    for j=1:4:48
        H(j) =h(j) -(2.5e-5)*(q(j)/2)^2;
        H(j+1)=h(j+1)-(1.652967e-5)*(q(j+1)/4)^2;
        H(j+2)=h(j+2)-(1.12889e-5)*(q(j+2)/4)^2;
        H(j+3)=h(j+3)-(1.736e-5)*(q(j+3)/6)^2;
    end

    %%%%%%%%%%%%%% 发电量计算，取效率为9.0
    for j=1:4:48 % 出力系数取为 8.8263，可以改进，求每个月出力
        Em((j-1)/4+1)=1003*(8.8263*H(j)*q(j)+8.8263*H(j+1)*q(j+1)+...
            8.8263*H(j+2)*q(j+2)+8.8263*H(j+3)*q(j+3))*t;
        Es(j) =1003*8.8263*H(j)*q(j)*t;
        Es(j+1)=1003*8.8263*H(j+1)*q(j+1)*t;
        Es(j+2)=1003*8.8263*H(j+2)*q(j+2)*t;
        Es(j+3)=1003*8.8263*H(j+3)*q(j+3)*t;
    end

    %%%%%%%%%%%%%% 梯级电站的总体年发电量
    result=0;
    for j=1:12
        result=result+Em(j); % 月电量累计求和
    end

```

%% 电量从焦耳换算成千
瓦时

```
result=result/3600000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 罚函数计算
G1=zeros(size(x));
G2=zeros(size(x));
for i=1:4:48    % 水轮机水头上界约束
    G1(i)= max([H(i)-3.4,0]);
    G1(i+1)=max([H(i+1)-8.8,0]);
    G1(i+2)=max([H(i+2)-26.7,0]);
    G1(i+3)=max([H(i+3)-8.04,0]);
end

for i=1:4:48    % 水轮机水头下界约束
    G2(i)= max([-H(i)+2,0]);
    G2(i+1)=max([-H(i+1)+6.8,0]);
    G2(i+2)=max([-H(i+2)+20,0]);
    G2(i+3)=max([-H(i+3)+4.16,0]);
end
G=[G1 G2];
n=length(find(G~=0)); % 违反约束的个数
violent=r*sum(G);    %输出违反约束的程度与个数
% 库水位的约束在主函数中通过设置粒子边界来实现
```

3 用于展示结果的程序 （记住还有 maple 的公式推导）

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 本函数用于展示优化调度过程
load data_month
load x_best
x=xm;
r=1e8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 将最优解带回目标函数进行计算
[result,violent,n,discharge,Em,Es]=target_maxf(x,r);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 展示年最大发电量
result=num2str(result);
disp(strcat('年发电量: ',result));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 展示优化调度方案(库容为决策变量
12*4)
%(编码方式) 渭沱->利泽->草街->井口
fprintf('最优调度方案:\n');
x=zeros(12,4);
for i=1:12
    for j=1:4
        x(i,j)=xm((i-1)*4+j);
    end
end
x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 展示每个电站每个月的弃水
d=discharge;
discharge=zeros(12,4);
fprintf('弃水情况:\n');
for i=1:12
    for j=1:4
        discharge(i,j)=d((i-1)*4+j);
    end
end
discharge
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 计算每个电站月出力
c=3600000;
es=Es/c;
Es=zeros(12,4);
fprintf('各电站月出力:\n');
for i=1:12
```



```

        for j=1:4
            Es(i,j)=es((i-1)*4+j);
        end
    end
    Es
    %%%%%%%%%%%%%%% 画图展示一年的出力过程（库容变化）
    figure('Color',[1 1 1]);
    plot(x(:,1),'b-*','linewidth',2,'markersize',8);
    hold on;
    plot(x(:,2),'r-x','linewidth',2,'markersize',8);
    plot(x(:,3),'k--o','linewidth',2,'markersize',8);
    plot(x(:,4),'m--+', 'linewidth',2,'markersize',8);
    xlabel('月份');
    ylabel('库容(立方米)');
    legend('渭沱','利泽','草街','井口');
    title('梯级水库时段库容变化曲线');
    hold off;
    %%%%%%%%%%%%%%% 画图展示每个月的发电量 和个电站
    的月发电量
    figure('Color',[1 1 1]);
    plot(Em/c,'b-', 'linewidth',2);
    hold on;
    plot(Es(:,1),'b-*','linewidth',2,'markersize',8);
    plot(Es(:,2),'r-x','linewidth',2,'markersize',8);
    plot(Es(:,3),'k--o','linewidth',2,'markersize',8);
    plot(Es(:,4),'m--+', 'linewidth',2,'markersize',8);
    xlabel('月份');
    ylabel('发电量(KW/H)')
    legend('月总发电量','渭沱月发电量','利泽月发电量','草街月发电量','井口月发电量');
    title('月发电量变化曲线');
    %%%%%%%%%%%%%%% 画图展示每个月的来水量和各电站的
    弃水量
    t=31*24*60*60;
    figure('Color',[1 1 1]);
    plot(weituo_month*t,'b', 'linewidth',2);
    hold on;
    plot(lize_month*t,'r', 'linewidth',2);
    plot(luoduxi_month*t,'k', 'linewidth',2);
    plot(discharge(:,1),'b-*','linewidth',2,'markersize',8);
    plot(discharge(:,2),'r-x','linewidth',2,'markersize',8);
    plot(discharge(:,3),'k--o','linewidth',2,'markersize',8);
    plot(discharge(:,4),'m--+', 'linewidth',2,'markersize',8);
    xlabel('月份');
    ylabel('水量(立方米)');

```

```
legend('渭沱来水量','利泽来水量','罗渡溪来水量','渭沱弃水量','利泽弃水量','草街弃水量','井  
口弃水量');  
title('来水量与弃水量曲线');  
hold off;
```