西安電子科技大學

题　　目：

学　　院：

专　　业：

姓　　名：

学　　号：

一．实验目的

罚函数法又称乘子法，是指将有约束最优化问题转化为求解无约束最优化问题：其中 M 为足够大的正数， 起"惩罚"作用， 称之为罚因子，F(x, M)称为罚函数。内部罚函数法也称为障碍罚函数法。

这种方法是在可行域内部进行搜索，约束边界起到类似围墙的作用，如果当前解远离约束边界时，则罚函数值是非常小的，否则罚函数值接近无穷大的方法。在进化计算中，研究者选择外部罚函数法的原因主要是该方法不需要提供初始可行解。其中 B(x)是优化过程中新的目标函数，Gi 和 Hj 分别是约束条件 gi(x)和 hj(x)的函数，ri 和 cj 是常数，称为罚因子。

用外点罚函数法和混合罚函数法编程求解

$$\min f(x) = -x_1 + x_2$$
$$s.t.\begin{cases} g_1(x) = \ln(x_2) \ge 0 \\ h_1(x) = x_1 + x_2 - 1 = 0 \end{cases} \tag{1}$$

终止限取 ε=10^-5。

用内点罚函数法编程求解

$$\min\ [\frac{1}{3}(x_1+1)^3 + x_2]$$
$$s.t.\begin{cases} x_1 - 1 \ge 0 \\ x_2 \ge 0 \end{cases} \tag{2}$$

取初始点 X0=[3,4]’，初始障碍因子 u=10，缩小系数取 c=0.1，终止限取 ε=10^-5。

二．实验环境

MATLAB 程序语言。

三．实验内容和步骤

外点罚函数法：

对于以上有约束最优化问题，构造一函数

$$F(X, M_k) = f(x) + M_K \alpha(X) \tag{3}$$

其中

$$\alpha(X) = \sum_{j=1}^{m}[h_j(X)]^2 + \sum_{i=1}^{l}[g_i(X)]^2 u(g_i(X)) \tag{4}$$

在式（4）中，α(X)又称为惩罚函数，

$$\alpha(X)\begin{cases}=0, X\in D\\ >0, X\notin D\end{cases} \tag{5}$$

$$u(g_i(X))=\begin{cases}0, g_i(X)\ge 0\\ 1, g_i(X)<0\end{cases} \tag{6}$$

Mk>0 是一个逐渐增大的参数，称为惩罚因子。F(X,Mk)又称为增广目标函数。

显然，增广目标函数 F(X,Mk)是定义在 Rn 上的一个无约束函数。由增广目标函数 F(X,Mk)的构造知当 X 属于 D 时，F(X,Mk)=f(X).此时 F(X,Mk)的最优解就是问题的最优解。

算法步骤：

1 选定初始点 X_0，初始惩罚因子 M_1 >0，惩罚因子放大系数 C=10，置 k=1。

2 假设已获迭代点 X_{k-1}，以 X_{k-1} 为初始点，求解无约束问题

$$\min F(X, M_k) \tag{7}$$

设其最优点为 X_k。

3 若 $M_k\alpha(X)\le\varepsilon$，则 X_k 就是所要求的最优解，打印 $(X_k, f(X_k))$，结束；否则转 4.

4 置 $M_{k+1}=CM_k$，k=k+1，转 2.

内点罚函数法：

内点罚函数法的思想为，在 D 的边界设置一道阻碍，当从可行域 D 中的某点 X0 出发进行迭代时，每当迭代点靠近 D 的边界时，便被此边界上的障碍阻挡碰回，这种阻挡实质是一种惩罚，，换句话说，所谓阻挡就是当迭代点靠近 D 的边界时，离边界越近函数值增加越大，特别是当迭代点到达边界上时，函数值变为无穷大。由此可以想象不可能在靠近 D 的边界上取得最优解，只能在远离 D 的边界内找到最优解。按照这种想法构造如下增广目标函数

$$F(X, r_k)=f(x)+r_K\sum_{i=1}^{l}\frac{1}{g_i(X)} \tag{8}$$

其中，r_k 称为障碍因子，称为 $\sum_{i=1}^{l}\frac{1}{g_i(X)}$ 障碍函数。

显然 F(X,rk)和 f(X)都定义在 D 内，rk 取值较小时，当迭代点远离边界，F 约等于 f(X)，此时 F 的最优解可作为问题的最优解；但当迭代点靠近 D 的边界时，由 F 可知，即使 rk 取值很小，F 的函数值变化很大此时显然不能在区域 D 的边界附近求得 F 的最优解，于是迫使迭代点被阻挡碰回到 D 的边界去寻优。当 rk 趋近 0 时，X 即为问题的最优解。

算法步骤：

1 选定初始点 $X_0 \in D$，初始障碍因子 r_1=10，障碍因子缩小系数 C<1，置 k=1。

2 假设已获迭代点 X_{k-1}，以 X_{k-1} 为初始点，求解无约束问题

$$\min F(X, r_k) \qquad \textbf{(9)}$$

设其最优点为 $X(r_k)$。

3 若 $r_k \sum_{i=1}^{l} \frac{1}{g_i(X(r_k))} \le \varepsilon$，则 $X(r_k)$ 就是问题的最优解，打印 $(X(r_k), f(X(r_k)))$，结束；否则转 4.

4 置 $r_{k+1} = Cr_k$，k=k+1，转 2.

混合罚函数法：

本文利用内点形式的混合罚函数，即

$$F(X, r_k) = f(x) + r_K \sum_{i=1}^{l} \frac{1}{g_i(X)} + \frac{1}{\sqrt{r_K}} \sum_{j=1}^{m} [h_j(X)]^2 \qquad (10)$$

初始点 X_0 应选为满足各个不等式约束条件的点，障碍因子 r_k 也应按照内点罚函数法取为递减序列，在 F 中，$\sum_{i=1}^{l} \frac{1}{g_i(X)}$ 的作用是限制搜索跑出不等式约束确定的区域，相当于内点罚函数法，而 $\sum_{j=1}^{m} [h_j(X)]^2$ 的作用是迫使搜索点向等式约束面靠近，相当于外点罚函数法。

算法步骤：

1 选定初始点 X_0，要求满足不等式约束，初始障碍因子 r_1，惩罚因子缩小系数 C<1，置 k=1。

2 假设已获迭代点 X_{k-1}，以 X_{k-1} 为初始点，求解无约束问题

$$\min F(X, r_k) \qquad （11）$$

设其最优点为 $X(r_k)$。

3 若 $\|X_k - X_{k-1}\| \le \varepsilon$，则 $X(r_k)$ 就是问题的最优解，打印 $(X(r_k), f(X(r_k)))$，结束；否则转 4.

4 置 $r_{k+1} = Cr_k$，k=k+1，转 2.

四．实验结果与分析

5.1 外点罚函数法

运行如下：

```
clear

clc

m=zeros(1,20);a=zeros(1,20);b=zeros(1,20);f0=zeros(1,20);

syms x1 x2 M f;

m(1)=1; %初始惩罚因子

c=10;%c 为递增系数。赋初值。

a(1)=10;b(1)=10;%初始点

f0(1)=0;

v = [x1,x2];

f=-x1+x2+M*((x1+x2-1)^2+(x2-1)^2);%外点罚函数

g = jacobian(f,v);%求梯度

Hesse = jacobian(g,v);%求 hesse 矩阵


%外点法 M 迭代循环

for k=1:100

    x1=a(k);x2=b(k);M=m(k);

    %牛顿法求最优值

    for n=1:100
```

```
        v = [x1,x2];
        G = eval(g);
        hesse = eval(Hesse);
        if(double(abs(G)<=1e-5))%最优值收敛条件
            a(k+1)=double(x1);
            b(k+1)=double(x2);
            f0(k+1)=double(eval(f));
            break;
        else
            v=v'-pinv(hesse)*G';
            x1=v(1,1);x2=v(2,1);
        end
    end
if(double(M*((x1+x2-1)^2+(x2-1)^2))<=1e-5) %惩罚因子迭代收敛条件
      fprintf('输出最优点坐标为(%f,%f)\n',a(k+1),b(k+1));
      fprintf('惩罚因子迭代次数为%d 次\n',k);
      fprintf('最终最优值为%f\n',f0(k+1));
      break;
else
      m(k+1)=c*m(k);
end
end
```

输出最优点坐标为(0.000001,0.999999)

惩罚因子迭代次数为 7 次

最终最优值为 0.999999

5.2 内点罚函数法

运行如下：

```
clear
```

```
clc
r=zeros(1,50);a=zeros(1,50);b=zeros(1,50);f0=zeros(1,50);
syms x1 x2 R f;
r(1)=10; %初始障碍因子
c=0.7;%c 为递减系数。赋初值。
a(1)=3;b(1)=4;%初始点
f0(1)=100;
v = [x1,x2];
f=((x1+1)^3)/3+x2+R*(1/(x1-1)+1/x2);%内点罚函数
g = jacobian(f,v);
Hesse = jacobian(g,v);

%外点法 M 迭代循环
for k=1:100
    x1=a(k);x2=b(k);R=r(k);
    %牛顿法求最优值
    for n=1:100
        v = [x1,x2];
        G = eval(g);
        hesse = eval(Hesse);
        if(double(abs(G))<=1e-5)%最优值收敛条件
            a(k+1)=double(x1);
            b(k+1)=double(x2);
            f0(k+1)=double(eval(f));
            break;
        else
            v=v'- pinv(hesse)*G';
            x1=v(1,1);x2=v(2,1);
        end
    end
```

```
if(double(R*(1/(x1-1)+1/x2))<=1e-5) %障碍因子迭代收敛条件

        %输出最优点坐标，障碍因子迭代次数，最优值

        fprintf('输出最优点坐标为(%f,%f)\n',a(k+1),b(k+1));

        fprintf('障碍因子迭代次数为%d 次\n',k);

        fprintf('最终最优值为%f\n',f0(k+1));

        break;

else

        r(k+1)=c*r(k);

end

end
```

```
输出最优点坐标为(1.000001,0.000003)

障碍因子迭代次数为 79 次

最终最优值为 2.666684
```

5.3 混合罚函数法

运行如下：

```
clear

clc

m=zeros(1,20);a=zeros(1,20);b=zeros(1,20);f0=zeros(1,20);

syms x1 x2 M f;

m(1)=1; %初始惩罚因子

c=10;%c 为递增系数。赋初值。

a(1)=10;b(1)=10;%初始点

f0(1)=0;

v = [x1,x2];

f=-x1+x2+M*((x1+x2-1)^2+(x2-1)^2);%外点罚函数

g = jacobian(f,v);%求梯度

Hesse = jacobian(g,v);%求 hesse 矩阵
```

```
%外点法 M 迭代循环
for k=1:100
    x1=a(k);x2=b(k);M=m(k);
    %牛顿法求最优值
    for n=1:100
        v = [x1,x2];
        G = eval(g);
        hesse = eval(Hesse);
        if(double(abs(G)<=1e-5))%最优值收敛条件
            a(k+1)=double(x1);
            b(k+1)=double(x2);
            f0(k+1)=double(eval(f));
            break;
        else
            v=v'-pinv(hesse)*G';
            x1=v(1,1);x2=v(2,1);
        end
    end
if(double(M*((x1+x2-1)^2+(x2-1)^2))<=1e-5) %惩罚因子迭代收敛条件
      fprintf('输出最优点坐标为(%f,%f)\n',a(k+1),b(k+1));
      fprintf('惩罚因子迭代次数为%d 次\n',k);
      fprintf('最终最优值为%f\n',f0(k+1));
      break;
else
      m(k+1)=c*m(k);
end
end
```

输出最优点坐标为(0.000001,0.999999)

惩罚因子迭代次数为 7 次

最终最优值为 0.999999

五．总结

通过此次上机实验，对内点罚函数法，外点罚函数法，混合法函数法有了有了更深入了解，在迭代求最优值的过程中使用了梯度下降和牛顿法。构造惩罚函数时可以用其他的不等式来替换对数不等式，牛顿法要设置好初始点，而且发现布长要在 **0.4-0.7** 之间。