数值最优化方法实验报告
约束问题求解

米科润　19 信计二班

201905755824

July 9, 2021

目录

1 外点法

求解问题

$$min(x_1-1)^2+(x_1-x_2)^2+(x_2-x_3)^2$$
$$s.t.x_1(1+{x_2}^2)+{x_3}^4-4-3\sqrt{2}$$
$$-10\leq x_i\leq 10, i=1,2,3$$
$$x_0=(2,2,2)^T$$

构建下述 Matlab 文件

- 目标函数 obj.m
- 约束条件函数 constrains.m
- 增广目标函数 Al obj.m
- 罚函数 compare.m
- 外点罚函数求解函数 Al main.m

```
%%
function f=obj(x)
f=(x(1)-1)^2+(x(1)-x(2))^2+(x(2)-x(3))^4;
end
%%
function [h,g]=constrains(x)
h=x(1)*(1+x(2)^2)+x(3)^4-4-3*sqrt(2);
g(1)=x(1)+10;
g(2)=x(2)+10;
g(3)=x(3)+10;
g(4)=-x(1)+10;
g(5)=-x(2)+10;
g(6)=-x(3)+10;
end
%%
function f=AL_obj(x)
global pena N_equ N_inequ;
h_equ=0;
h_inequ=0;
[h,g]=constrains(x);
for i=1:N_equ
```

```
22          h_equ=h_equ+h(i).^2;
23      end
24      for i=1:N_inequ
25          h_inequ=h_inequ+(min(g(i),0)).^2;
26      end
27      f=obj(x)+pena*(h_equ+h_inequ);
28      end
29      %%
30      function f=compare(x)
31      global pena N_equ N_inequ;
32      h_equ=0;
33      h_inequ=0;
34      [h,g]=constrains(x);
35      for i=1:N_equ
36          h_equ=h_equ+h(i).^2;
37      end
38      for i=1:N_inequ
39          h_inequ=h_inequ+(min(g(i),0)).^2;
40      end
41      f=pena*(h_equ+h_inequ);
42      end
43      %%
44      function [X,FVAL]=AL_main(x_al,N_equ,N_inequ)
45      global pena N_equ N_inequ;
46      pena=0.1;
47      c_scale=2;
48      e_al=1e-6;
49      max_itera=100;
50      out_itera=1;
51      while out_itera<max_itera
52          x_al0=x_al;
53          [X,FVAL]=fminunc(@AL_obj,x_al0);
54          x_al=X;
55          if compare(x_al)≤e_al
56              break;
57          end
58          pena=c_scale*pena;
59          out_itera=out_itera+1;
60      end
61      X=x_al;
```

```
62        FVAL=obj(X);
63        end
```

构建 main.m 输入参数求解

```
1         function main()
2         clc;clear;
3         x_al=[2,2,2];
4         N_equ=1;
5         N_inequ=6;
6         [X,FVAL]=AL_main(x_al,N_equ,N_inequ);
7         disp(X);disp(FVAL);
8         end
```

结果如下

```
<stopping criteria details>
迭代结果为：
   1.104795485345797    1.196572225359752    1.535284306381924

   0.032567089882592
```

图 1: 外点罚函数法结果

2 内点法

求解问题

$$minf(x) = -x_1x_2x_3$$
$$s.t. - {x_1}^2 - 2{x_2}^2 - 4{x_3}^2 + 48 \geq 0$$
$$x_0 = (1, 1, 1)$$

构建下述 Matlab 文件

- 目标函数 obj.m
- 约束条件函数 constrains.m
- 增广目标函数 Al obj.m
- 罚函数 compare.m

- 外点罚函数求解函数 Al main.m

```
%%
function f=obj(x)
f=-(x(1)*x(2)*x(3));;
end
%%
function [h,g]=constrains(x)
g=-x(1)^2-2*x(2)^2-4*x(3)^2+48;
end
%%
function f=AL_obj(x)
global pena N_inequ;
h_inequ=0;
g=constrains(x);
for i=1:N_inequ
    h_inequ=h_inequ-log(g(i));
end
f=obj(x)+pena*(h_inequ);
end
%%
function f=compare(x)
global pena N_inequ;
h_inequ=0;
g=constrains(x);
for i=1:N_inequ
    h_inequ=h_inequ-log(g(i));
end
f=pena*(h_inequ);
end
%%
function [X,FVAL]=AL_main(x_al,N_equ,N_inequ)
global pena N_equ N_inequ;
pena=10;
c_scale=0.5;
e_al=1e-6;
max_itera=100;
out_itera=1;
while out_itera<max_itera
```

```
        x_al0=x_al;
        [X,FVAL]=fminunc(@AL_obj,x_al0);
        x_al=X;
        if compare(x_al)≤e_al
            break;
        end
        pena=c_scale*pena;
        out_itera=out_itera+1;
    end
    X=x_al;
    FVAL=obj(X);
    end
```

问题：只有初始点非常靠近最优解时，所得结果近似正确

构建 main.m 输入参数求解

```
function main()
clc,clear;
x_al=[4,2*sqrt(2),2];
N_inequ=1;
[X,FVAL]=AL_main(x_al,N_inequ);
disp(X);disp(FVAL);
end
```

结果如下

```
迭代结果为：
   4.100000000000000   2.863782463805518   2.025000000000000

-23.776553905745310
```

图 2: 内点罚函数法结果

3 乘子法

求解问题

$$minf(x) = -3{x_1}^2 - {x_2}^2 - 2{x_3}^2$$
$$s.t.{x_1}^2 + {x_2}^2 + {x_3}^2 = 0$$
$$x_2 \geq x_1$$
$$x_1 \geq 0$$
$$x_0 = (0,0,0)^T$$

构建下述 Matlab 文件

- PHR 算法函数 multphr.m
- 增广拉格朗日函数 mpsi.m
- 增广拉格朗日函数的梯度 dmpsi.m
- 目标函数 f1.m
- 等式约束函数 h1.m
- 不等式约束函数 g1.m
- 目标函数的梯度 df1.m
- 等式约束的 Jacob 矩阵转置 dh1.m
- 不等式约束的 Jacob 矩阵转置 dg1.m

```
1          %%
2          function ...
               [x,mu,lam,output]=multphr(fun,hf,gf,dfun,dhf,dgf,x0)
3          theta=0.8;   eta=2.0;
4          k=0;   ink=0;
5          epsilon=1e-5;
6          x=x0;   he=feval(hf,x);   gi=feval(gf,x);
7          n=length(x);   l=length(he);   m=length(gi)
8          mu=0.1*ones(l,1);   lam=0.1*ones(m,1);
9          betak=10;   betaold=10;
10         while(betak>epsilon & k<maxk)
11         [ik,x,val]=bfgs('mpsi','dmpsi',x0,fun,hf,gf,dfun,dhf,dgf,...
12         mu,lam,sigma);
13         ink=ink+ik;
```

```
14          he=feval(hf,x); gi=feval(gf,x);
15          betak=sqrt(norm(he,2)^2+norm(min(gi,lam/sigma),2)^2);
16          if betak>epsilon
17              mu=mu-sigma*he;
18              lam=max(0.0,lam-sigma*gi);
19                  if(k>=2 & betak>theta*betaold)
20                      sigma=eta*sigma;
21                  end
22          end
23          k=k+1;
24          betaold=betak;
25          x0=x;
26          end
27          f=feval(fun,x);
28          output.fval=f;
29          output.iter=k;
30          output.inner_iter=ink;
31          output.beta=betak;
32          %%
33          function psi=mpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lam,sigma)
34          f=feval(fun,x);  he=feval(hf,x);  gi=feval(gf,x);
35          l=length(he);  m=length(gi);
36          psi=f;  s1=0.0;
37          for(i=1:l)
38              psi=psi-he(i)*mu(i);
39              s1=s1+he(i)^2;
40          end
41          psi=psi+0.5*sigma*s1;
42          s2=0.0;
43          for(i=1:m)
44              s3=max(0.0,lam(i)-sigma*gi(i));
45              s2=s2+s3^2-lam(i)^2;
46          end
47          psi=psi+s2/(2.0*sigma);
48          %%
49          function dpsi=dmpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lam,sigma)
50          dpsi=feval(dfun,x);
51          he=feval(hf,x);  gi=feval(gf,x);
52          dhe=feval(dhf,x);  dgi=feval(dgf,x);
53          l=length(he);  m=length(gi);
```

```
54          for(i=1:l)
55              dpsi=dpsi+(sigma*he(i)-mu(i))*dhe(:,i);
56          end
57          for(i=1:m)
58              dpsi=dpsi+(sigma*gi(i)-lam(i))*dgi(:,i);
59          end
60          %%
61          function f=f1(x)
62          f=-3*x(1)^2-x(2)^2-2*x(3)^2;
63          end
64          %%
65          function he=h1(x)
66          he=x(1)^2+x(2)^2+x(3)^2-3;
67          end
68          %%
69          function gi=g1(x)
70          gi=zeros(2,1);
71          gi(1)=-x(1)+x(2);
72          gi(2)=x(1);
73          end
74          %%
75          function g=df1(x)
76          g=[-6*x(1);-2*x(2);-4*x(3)];
77          end
78          %%
79          function dhe=dh1(x)
80          dhe=[2*x(1), 2*x(2), 2*x(3)]';
81          end
82          %%
83          function dgi=dg1(x)
84          dgi=[-1   1; 1  0; 0  0];
85          end
```

命令窗口输入:

```
1       x0=[0,0,0]';
2       [x,mu,lam,output]=multphr('f1','h1','g1','df1','dh1','dg1',x0)
```

结果如下

```
>>  x0=[0,0,0]';
    [x,mu,lam,output]=multphr('f1','h1','g1','df1','dh1','dg1',x0)

x =

   1.224746061229717
   1.224744557859691
                   0


mu =

  -1.499996036173343


lam =

   1.224742300283214
                   0


output =

  包含以下字段的 struct:

         fval: -6.000007975500043
         iter: 7
    inner_iter: 50
         beta: 2.620611298250149e-06
```

图 3: 乘子法结果